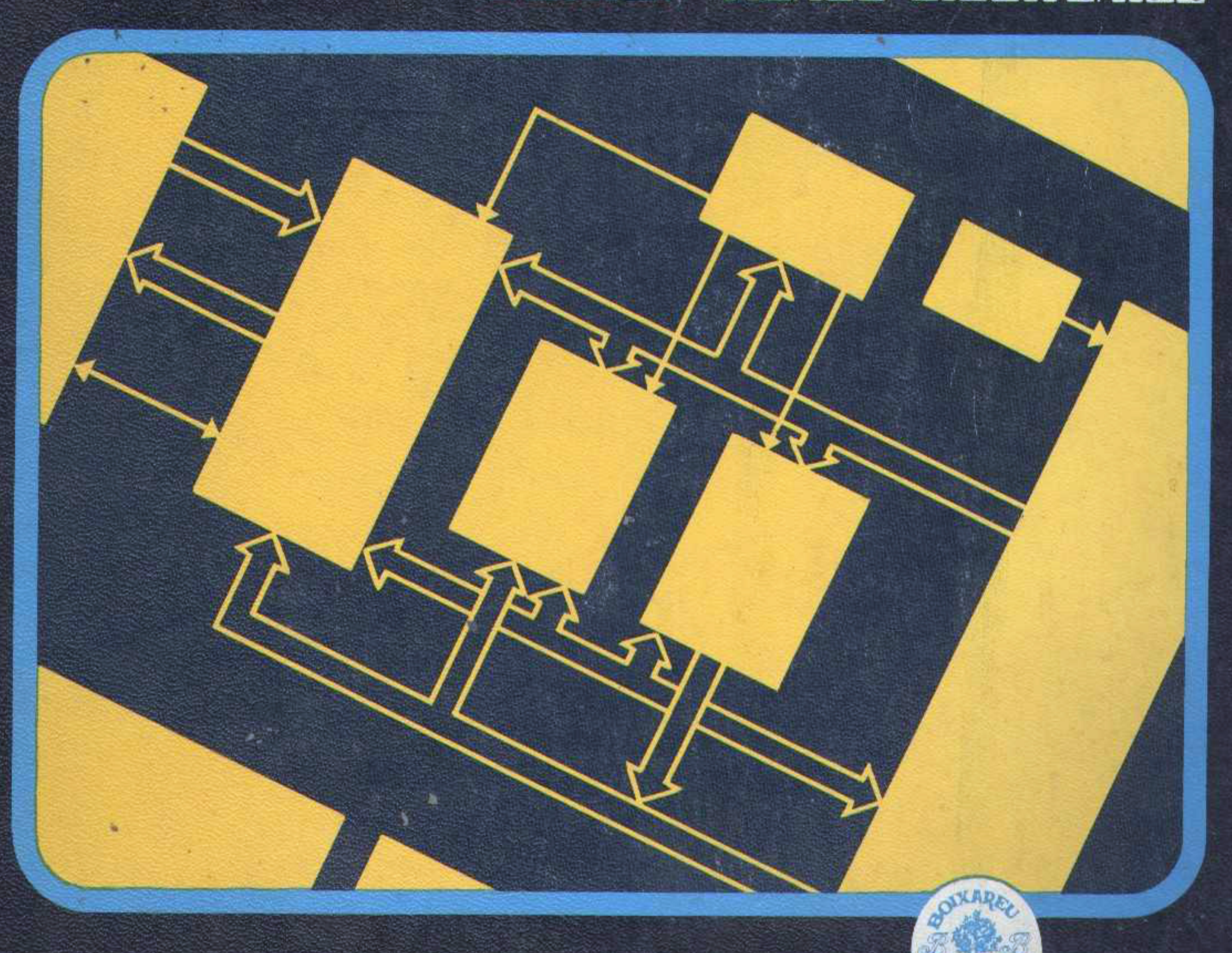
INTERCONEXION DE PERIFERICOS A MICROPROCESADORES

3ª edición

SERIE: mundo electrónico



BOIXAREU EDITORES

marcombo
BOIXAREU EDITORES

interconexión de periféricos a microprocesadores

SERIE: mundo electrónico

interconexión de periféricos a microprocesadores

por varios autores bajo la coordinación de José Mompín Poblet, director de la revista "Mundo Electrónico"



marcombo
BOIXAREU EDITORES

BARCELONA - MEXICO

Gran Via de les Corts Catalanes, 594
08007 Barcelona

3.ª edición

ISBN: 84-267-0654-1 rústica
ISBN: 84-267-0655-X encuadernado
Depósito Legal B-25954 - 1987
Impreso en España
Printed in Spain
Fotocomposición: FOINSA - Gran Via de les Corts Catalanes, 569 - 08011 Barcelona
Impresión: GERSA, Indust. Gráfica - Tambor del Bruc, 6 - 08970 Sant Joan Despí (Barcelona)

índice

prólogo

Tras el éxito alcanzado por el libro «Microprocesadores y Microcomputadores» de la Serie «MUNDO ELECTRÓNICO», del que ya se han impreso varias ediciones, nos pareció interesante y necesario confeccionar otro sobre el tema de los periféricos y sus interfaces.

La penetración del microprocesador en el mercado español a todos los niveles y en todas las áreas es una realidad. Nuestros jóvenes estudiantes de ingeniería, ingeniería técnica y FP2 salen de sus respectivas escuelas con un considerable bagaje técnico, en el que el tema de la Microinformática ocupa un lugar destacado; en otros casos, especialmente a niveles medios (ingeniería técnica y FP2) más que un extenso conocimiento del μP/μC se les inculca la inquietud por el tema, de manera que el postgraduado se siente en la necesidad de profundizar en la materia y actualizar sus conocimientos constantemente.

«Interconexión de periféricos a microprocesadores» *es el primer libro en lengua española que versa sobre esta materia específica, y viene a completar el tema afrontado por el primer libro de nuestra Serie. A pesar de la aridez del tema, los diferentes coautores han puesto una atención especial en tratar sus respectivos capítulos con un estilo claro y didáctico.*

Otra de las ventajas que creemos que presentan los distintos libros de la Serie MUNDO ELECTRÓNICO es su actualización, ya que debido a la técnica utilizada, son confeccionados en un tiempo récord y siempre al día en cada una de las siguientes ediciones. Precisamente esta es la novedad que aporta esta tercera edición, en la que se han revisado varios capítulos, que con el transcurso del tiempo habían quedado en parte obsoletos.

Con esta nueva edición revisada y actualizada, esperamos aportar nuestro granito de arena a la formación de tantos y tantos estudiantes y técnicos como existen en el país, conscientes de la vertiginosa evolución tecnológica en la que nuestro sector se halla inmerso, y a la que no podemos sustraernos.

José Mompín Poblet

conceptos generales 1

Ramón Pericas

1.1 GENERALIDADES

Desde la aparición del primer microprocesador, el INTEL 4004 en 1971, el universo de los microcomputadores, así como los campos de aplicación en los que se utiliza, se han ido ampliando con gran rapidez.

La introducción del microprocesador ha significado una verdadera revolución en el campo del diseño industrial, y es de prever un mayor impacto todavía con el desarrollo de nuevos sistemas más potentes y la reducción de costes de fabricación debido a la rápida evolución tecnológica en este campo.

El microprocesador aparece en la cuarta generación de ordenadores, marcada por la utilización de tecnologías de alta escala de integración (LSI), que ha hecho posible el desarrollo de los circuitos integrados que componen un sistema microcomputador:

— microprocesadores (autómatas programables complejos);
— memorias de semiconductores de gran capacidad y bajo coste;
— circuitos de soporte (interfaces, controladores, etc.), compactos y versátiles.

La utilización de sistemas basados en microcomputadores desplaza a la lógica cableada y a sistemas programables más complejos hacia otros márgenes de utilización, a la vez que posibilita la introducción de la electrónica en general, y la informática en particular, en nuevos campos de aplicación.

Respecto a la sustitución de sistemas de lógica cableada por microcomputadores, éste presenta las siguientes ventajas:

a) El microprocesador puede reemplazar un número elevado de circuitos integrados, lo cual presenta ventajas en cuanto a facilidad y modularidad del diseño, reducción de costes, aumento de fiabilidad por el menor número de elementos e interconexiones y disminución del consumo.
b) El sistema resultante es más flexible, por lo que permite más opciones y/o modificaciones a la vez que posibilita futuras expansiones del sistema.
c) Se puede trabajar con estructuras de información más desarrolladas, las operaciones aritméticas y lógicas pueden ser más complejas y sobre palabras de longitud variable, y se aumenta la capacidad de decisión del sistema.
d) Se posibilita el soporte de un sistema de entradas y salidas potente en cuanto a número y diversificación de accesos.
e) Se simplifica la diagnosis de fallos, las reparaciones y el mantenimiento.
f) Permite concebir sistemas de una forma modular y estructurada.
g) Se reduce el período de desarrollo y comercialización del equipo.

Los miniordenadores son máquinas aptas para la utilización de carácter general, aunque muchos son utilizados en aplicaciones específicas, por lo que precisan ser rápidos, de gran capacidad y fácilmente reprogramables, y tanto más cuanto más se sube en la escala de potencias de ordenadores. Frente a éstos, el microcomputador encuentra su campo de utilización más característico en aplicaciones donde la potencia de proceso requerida es más moderada y, normalmente, son aplicaciones particulares, es decir, con programa fijo para una función específica.

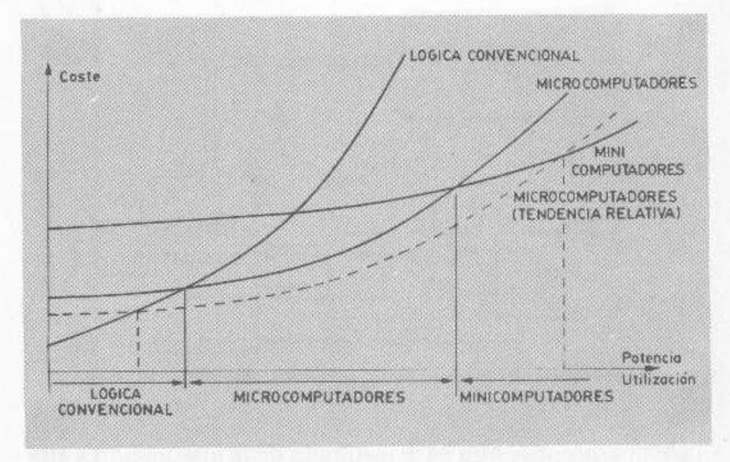


Figura 1.1 Margen de utilización de los microcomputadores.

La frontera entre micro y minicomputadores se va haciendo cada vez más ambigua debido al desarrollo de microcomputadores cada vez más potentes (TMS 9900, MCS 8086, M68000, Z-8000...), la incorporación de la tecnología LSI al desarrollo de minicomputadores (a partir de procesadores LSI trabajando en paralelo) y la sustitución de minicomputadores anteriores por microcomputadores integrados con software compatible con sus predecesores.

En la figura 1.1 se muestra, de forma relativa, las zonas de utilización de sistemas basados en lógica cableada, microprocesadores y minicomputadores.

Actualmente las aplicaciones de los microcomputadores cubren diversos sectores, entre los que cabría destacar:

a) Controladores programables: automatismos industriales, máquinas herramienta, etc.
b) Instrumentación: terminales interactivos, analizadores lógicos, «data-loggers», instrumentación biomédica, etc.
c) Controladores de periféricos: unidades de discos, control de impresoras, etc.
d) Control de procesos: sistemas de supervisión y control, etc.
e) Unidades de proceso aritmético y contables: ordenadores de gestión, cajas registradoras, etc.
f) Sistemas para la comunicación: transmisores y receptores de datos, etc.

1.2 ESTRUCTURA DE LOS MICROCOMPUTADORES

1.2.1 Organización de los microcomputadores

Los distintos grupos funcionales, o subsistemas, que debe poseer un ordenador se pueden resumir en los siguientes:

a) Un sistema de entradas y salidas para establecer la comunicación con el mundo exterior (recepción de órdenes y datos, y entrega de resultados).
b) Un sistema de memoria para la obtención de instrucciones (programa) y almacenamiento de información (datos).
c) Una unidad de cálculo para la ejecución de operaciones aritméticas y lógicas.
d) Una unidad de control capaz de coordinar el funcionamiento de todo el sistema y de tomar decisiones en función de resultados previos.

En la figura 1.2 se presenta en forma de diagrama de bloques la arquitectura típica de un ordenador de utilización general.

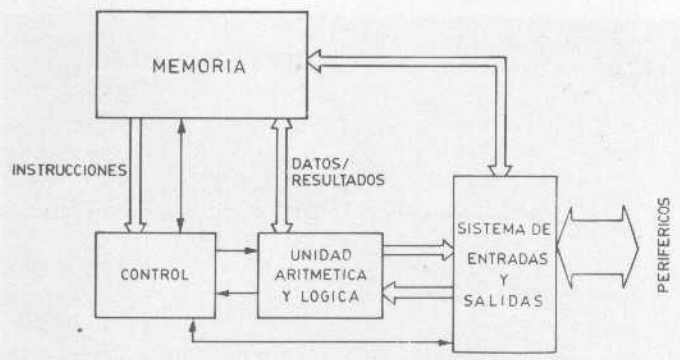


Figura 1.2 Diagrama de bloques de un ordenador de utilización general.

Un microprocesador es el resultado de la integración a alta escala de la unidad aritmética y lógica (ALU) y el subsistema de control, o gran parte del mismo, en una única unidad central de proceso (CPU). Un microcomputador es el ordenador completo que utiliza un microprocesador. En algunos casos, este elemento base contendrá también parte de otros subsistemas, como líneas de entrada y salida y/o memoria de programa o de datos, siendo el caso límite la

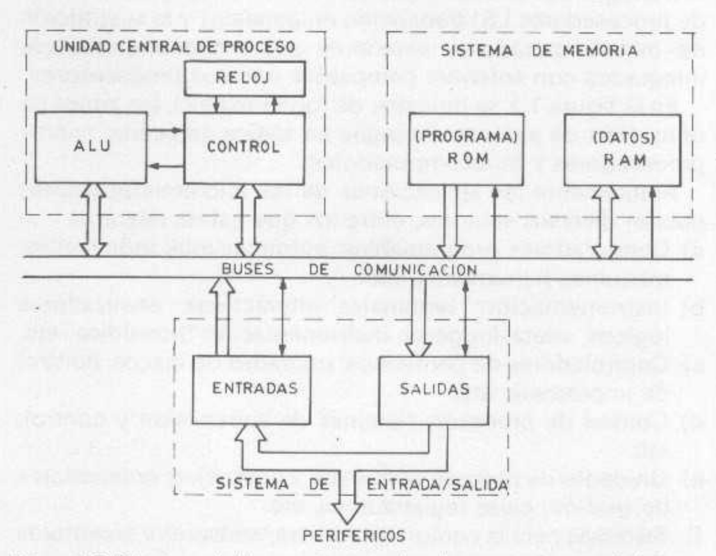


Figura 1.3 Diagrama de bloques de un ordenador con estructura de buses.

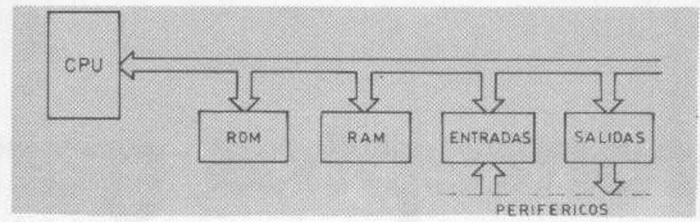


Figura 1.4. Estructura ortogonal.

integración de un microcomputador completo en un solo circuito LSI.

La arquitectura típica de los sistemas microcomputadores se presenta en la figura 1.3, en la que se muestran los distintos subsistemas enlazados entre sí por unas únicas vías de comunicación, agrupadas funcionalmente en buses o barras de comunicación.

Existen dos alternativas límites en cuanto a la organización concreta de un microcomputador:

1) *Estructura ortogonal:* partición del sistema en subsistemas disjuntos donde cada uno tiene una función específica. De esta forma es posible aumentar la capacidad de cada subsistema (ROM, RAM, accesos de entrada o de salida) de una forma independiente. La figura 1.4 muestra este tipo de estructura.

2) *Estructura empaquetada (packed):* integración de varias funciones distintas en un mismo circuito integrado. Existen circuitos integrados con dos o más funciones: ROM, RAM, accesos de entrada y salida programables, temporizadores, circuitos de control de interrupciones, etc. De esta forma el crecimiento de cada subsistema no es independiente. La figura 1.5 muestra este tipo de estructura.

En la práctica los microcomputadores tienen una estructura mixta, tendiendo hacia la estructura ortogonal los sistemas más complejos, con requerimientos de gran capacidad de memoria y/o accesos periféricos, y hacia una estructura más empaquetada los sistemas para aplicaciones menos complejas, por la ventaja que supone en la reducción del número de circuitos integrados.

En el desarrollo de distintas familias de microprocesadores y circuitos de soporte compatibles, se han seguido tendencias diversas, condicionando en gran parte la estructura general del microcomputador. Entre estas tendencias, no todas incompatibles, cabe destacar:

a) Incorporación del máximo número de funciones en un circuito dando lugar a microcomputadores integrados (Intel 8048, Texas 9940, Mostek 3870, AMI 2000).
b) Desarrollo de una CPU de forma que facilite la interconexión con circuitos estándar de memoria y de entrada/salida (Intel 8080/85, Motorola 6800, Zilog Z-80).
c) Posibilidad de una capacidad mínima formando un microcomputador con pocos circuitos integrados, logrando una configuración fácilmente ampliable (Rockwell PPS-8, Intel 8085).

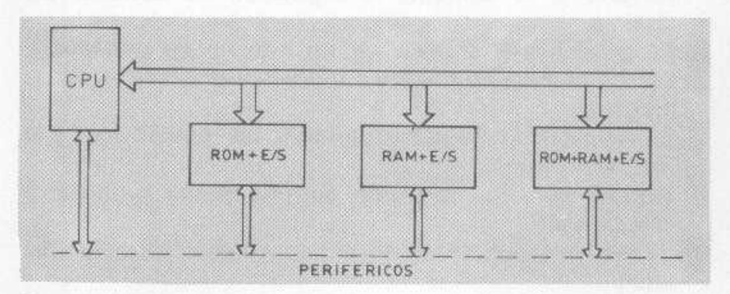


Figura 1.5 Estructura empaquetada.

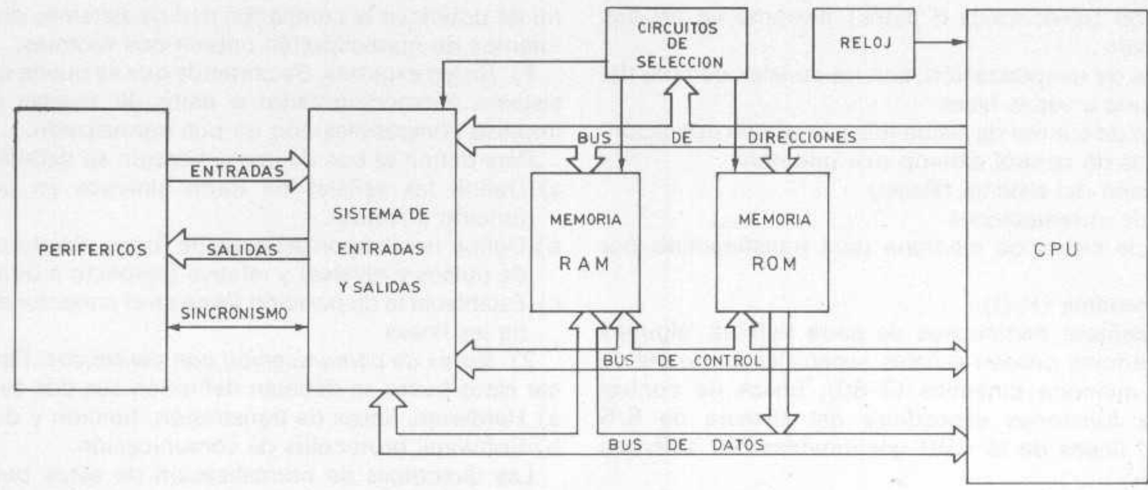


Figura 1.6 Interconexión de los subsistemas del microcomputador por medio de buses compartidos.

d) Estructura memoria-memoria con pocos registros internos, a fin de potenciar funciones propias de la CPU y permitir un servicio eficiente de interrupciones (Texas TMS 9900).
e) Optimización para funciones de controlador especializado, más que para formar un computador versátil de utilización general (Fairchild F-8), National SC/MP, Scientific Microsystems SMS 300).
f) Partición de la unidad aritmética en «rebanadas» (bit-slice) para conseguir longitudes de palabra variable (AMD 2900, Intel 3002, Motorola 10800, Fairchild 9400, Texas SBP 0400, Monolithic Memories 6701).
g) Posibilidad de microprogramación para lograr un repertorio de instrucciones arbitrario (Texas TMS1000, General Instruments CP 1600, National IMP-16, microprocesadores a rebanadas).

1.2.2 Estructura de buses

Las necesidades de comunicación en un sistema microcomputador aparecen a tres niveles distintos:

1) Comunicación entre los elementos constituyentes de un circuito integrado. Este sistema de comunicación constituye los buses internos, de estructura más o menos distinta para cada CPU considerada.

2) Comunicación entre los distintos subsistemas del microcomputador: unidad central de proceso, memoria y entrada/salida. El conjunto de las vías para este tipo de comunicación forman los buses externos, en general de carácter síncrono.

3) Comunicación con periféricos: esta comunicación es a nivel de sistema de entrada/salida con periférico. El conjunto de líneas de transmisión forman los buses periféricos, en general de carácter asíncrono. Estos buses y sus protocolos de comunicación se tratarán en otros capítulos.

La información desde o hacia la CPU y otros subsistemas puede ser de tres tipos: direcciones, datos y control; por lo que funcionalmente se encontrarán tres tipos de buses en el microcomputador, aunque no estén físicamente siempre separados, ya que pueden estar multiplexados en el tiempo sobre líneas comunes. La interconexión de la CPU con el sistema de memoria y E/S se realiza mediante estos buses, tal como se ilustra en la figura 1.6.

1) *Bus de direcciones:* son líneas de la CPU a memoria y sistema de entradas/salidas que sirven para enviar la dirección del elemento seleccionado para una transferencia. Estas líneas serán siempre gobernadas por la CPU, salvo en el caso de transferencias por acceso directo a memoria. Pueden compartir total o parcialmente las líneas físicas con el bus de datos.

2) *Bus de datos:* son líneas bidireccionales (o dos buses unidireccionales independientes) para el envío de información (instrucciones y datos) entre subsistemas. Su anchura, en general, es la longitud de palabra del microprocesador.

3) *Bus de control:* son líneas de entrada o salida de la CPU de órdenes o de información de estado que permiten coordinar la operación de todo el sistema.

El bus de control presenta variaciones notables de un microprocesador a otro. Existen dos tendencias en cuanto a su realización física:

a) Dar unas señales de control elementales, que pueden estar codificadas, y que deben ser interpretadas temporalmente respecto al estado en que se producen en cada ciclo de máquina. Esta estructura se observa principalmente en microprocesadores de las primeras generaciones.
b) Dar un conjunto de señales completo y demultiplexado a las que las memorias y periféricos responden de una forma pasiva, por lo que no son precisos circuitos especializados para la decodificación de las señales de control.

Las distintas líneas del bus de control se pueden agrupar funcionalmente en varios tipos distintos, dependiendo de cada microprocesador en particular el que se disponga de unas y otras funciones, así como la forma de realizarlas:

1) Señales de sincronización de transferencias: gobiernan el intercambio de información entre subsistemas. Estas señales son las de validación de lectura o escritura de memoria y E/S. En caso de que el microprocesador permita un funcionamiento asíncrono de los buses, se dispondrá de líneas de control (Ready, Wait,.) que permiten retrasar los ciclos de lectura y escritura en caso de que la memoria o periféricos tengan un tiempo de acceso superior al preciso para una transferencia síncrona gobernada sólo por la CPU.

2) Señales de estado. Salidas de la CPU que pueden significar dos tipos de información:

a) estado del microprocesador: tipo de ciclo de máquina en curso, inicio de ciclo (sincronismo), estado del sistema de interrupciones, inicialización...
b) estado de los buses: pueden indicar que los buses están en estado de alta impedancia, si el bus de datos está operando en modo de entrada o de salida, el tipo de

información (direcciones o datos) presente en un bus multiplexado...

3) Señales de temporización: son las señales de reloj del sistema, de una o varias fases.

4) Señales de control de secuencia o modo de operación. Son primitivas de control externo que originan:

a) Inicialización del sistema (Reset)
b) Petición de interrupciones
c) Petición de ciclos de memoria para transferencias por DMA
d) Paro del sistema (Halt)

5) Otras señales particulares de cada sistema: algunos microprocesadores poseen señales especiales como las de refresco de memoria dinámica (Z-80), líneas de control dedicadas a funciones específicas del sistema de E/S (COSMAC), líneas de la CPU gestionables por software (8085), etc.

En la figura 1.7 se muestran, a título de ejemplo, los buses del microprocesador Intel 8085.

En general, la estructura, respecto a si los buses son o no multiplexados, depende del microprocesador utilizado, aunque se observa una cierta evolución desde los primeros microprocesadores comercializados:

1) En los microprocesadores de la 1.ª generación, la limitación en el número de patillas (pins) de los circuitos integrados impone un multiplexaje de direcciones, datos y señales de control sobre unas mismas vías de comunicación, lo cual implica una gran complejidad en los sistemas externos a la CPU (registros, demultiplexores, decodificadores, temporizadores, etc.).

2) En la segunda generación se tiende a separar los buses por funciones, pudiéndose encontrar sistemas totalmente demultiplexados (6800) y otros en los que es preciso un sistema externo a fin de disponer de las señales de control de una forma operativa (8080).

3) En los microprocesadores más modernos se tiende a demultiplexar totalmente las señales de control de forma que se simplifique la interconexión con otros sistemas, y además se dispone de más líneas de control y de interrupciones, lo cual implica una vuelta a un sistema multiplexado de datos y direcciones, parcial (8085) o total en los microprocesadores de 16 bits (excepto el 9900 que mantiene los buses demultiplexados aumentando el número de terminales [de 40 a 64]).

Existe una tendencia hacia la normalización de los buses a

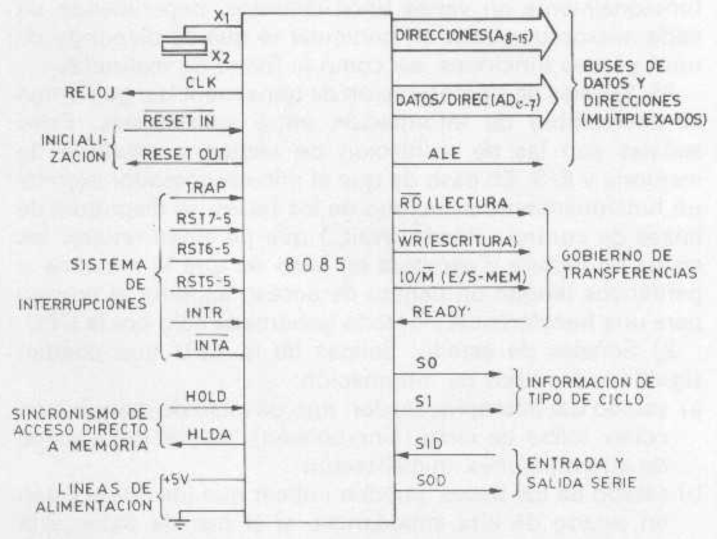


Figura 1.7 Terminales del microprocesador 8085.

fin de potenciar la compatibilidad de sistemas diversos. Los intentos de normalización cubren dos sectores:

1) *Buses externos*. Se pretende que se pueda construir un sistema microcomputador a partir de tarjetas de circuito impreso compatibles con un bus normalizado.

Para definir el bus de comunicación se deberá:

a) Definir las señales de forma unívoca en cuanto a la función a realizar.
b) Definir las temporizaciones de forma absoluta (duración de pulsos y niveles) y relativa (respecto a otras señales).
c) Establecer la disposición física en el conector de cada una de las líneas.

2) *Buses de comunicación con periféricos*. Para especificar estos buses se deberán definir en sus dos aspectos:

a) Hardware: líneas de transmisión, función y disposición.
b) Software: protocolos de comunicación.

Las directrices de normalización de estos buses cubren tres niveles distintos:

a) transmisión en paralelo con periféricos cercanos al microcomputador;
b) transmisión en serie con periféricos situados a una distancia mayor que justifique el intento de reducir el número de líneas;
c) unión a redes de periféricos y/o redes de ordenadores.

En posteriores capítulos se tratarán diversos buses y protocolos normalizados o muy generalizados en cuanto a utilización.

1.2.3 Unidad central de proceso

Aunque el interés de este libro se centra en la estructura y gestión de los sistemas de entrada y salida, se consideran en este apartado y el siguiente algunos aspectos de la unidad central de proceso y del sistema de memoria.

1.2.3.1 *Funciones*

La unidad central de proceso (CPU) es el sistema principal de un microcomputador, ya que es el elemento capaz de interpretar las intrucciones y coordinar su ejecución. La CPU está constituida por tres subsistemas funcionales:

1) *Unidad aritmética y lógica (ALU):* es el elemento calculador del sistema, capaz de realizar operaciones lógicas (AND, OR, XOR...) y aritméticas (suma, resta...).

2) *Unidad de control (UC):* es el elemento controlador del flujo de información (instrucciones y datos) en el sistema. Las funciones de la unidad de control son:

a) búsqueda de instrucciones en memoria;
b) decodificación, interpretación y ejecución de las instrucciones;
c) control de la secuencia de ejecución;
d) reconocimiento de primitivas externas de control: interrupciones, petición de acceso directo a memoria, órdenes de paro, espera, inicialización, etc.

3) *Generador de reloj*.

Estos tres subsistemas están total o parcialmente integrados en un único circuito LSI, el que se denomina microprocesador.

1.2.3.2 *Instrucciones*

El repertorio de instrucciones es muy variable en función del microprocesador utilizado. Existe una gran diferencia entre los primeros microprocesadores con un conjunto de instrucciones muy limitado, y los modernos, con un

conjunto de instrucciones potentes, cercano o superior al de muchos minicomputadores.

Por la función realizada, las instrucciones típicas de un microprocesador se pueden dividir en varias categorías:

1) *Instrucciones de transferencia:* movimiento de datos entre registros. Los registros fuente y destino de la transferencia pueden pertenecer a la CPU (acumulador y registros auxiliares), a memoria o al sistema de E/S.

2) *Instrucciones operativas:* ejecución de operaciones lógicas o aritméticas entre dos operandos (suma, resta, multiplicación...), sobre un solo operando (incremento, complemento, borrado...) o bit a bit (Y, O, O exclusiva, puesta a uno, complemento...).

3) *Instrucciones de control de secuencia:* saltos de programa y llamadas y retornos de subrutinas, incondicionales o condicionales.

4) *Instrucciones de control:* instrucciones que afectan al registro de estado, sistema de interrupciones, cambio de bancos de registros, paro del sistema, instrucciones especiales de E/S, etc.

Algunos sistemas son microprogramables, lo cual posibilita un repertorio de instrucciones arbitrario.

Existen distintas formas de direccionar los registros, palabras de memoria y accesos de entrada y salida que intervienen en la ejecución de una instrucción. Las posibilidades de direccionamiento de un microprocesador son importantes de tener en cuenta por su repercusión en la velocidad de ejecución, flexibilidad de la programación y necesidades en la capacidad de memoria.

Una instrucción se puede considerar dividida en dos partes: código de operación y operando (si es necesario). El operando, si existe, ocupa palabra(s) sucesiva(s) al código de operación. La información de la dirección se puede presentar de distintas formas:

a) *Direccionamiento implícito:* cuando el mismo código de operación indica la dirección del operando. Por la limitación que este método supone, en cuanto a capacidad, su utilización es restringida al direccionamiento de registros internos de la CPU y casos especiales de control de secuencia.

b) *Direccionamiento inmediato:* cuando el operando no representa una dirección sino que es uno de los datos de la operación.

c) *Direccionamiento directo:* cuando se indica de una forma absoluta la dirección efectiva del operando o la dirección de salto en caso de instrucciones de control de secuencia.

d) *Direccionamiento indirecto:* cuando el operando indica una dirección donde está la dirección efectiva del operando. Puede ser posible un encadenamiento de indirecciones hasta llegar a la dirección efectiva. La indirección puede ser respecto a un registro interno, punteros de memoria o de stack, o respecto a memoria, según donde esté el puntero de la dirección efectiva.

e) *Direccionamiento relativo al contador de programa:* el operando indica un desplazamiento, positivo o negativo, respecto a la dirección de la instrucción en curso.

f) *Direccionamiento indexado:* el operando indica un desplazamiento respecto a un puntero de memoria en la CPU (registro de índice).

g) *Direccionamiento por páginas:* cuando el operando indica la dirección efectiva en la página (memoria dividida en módulos de 256 palabras) en curso o respecto a la página cero (posiciones 0 a 255).

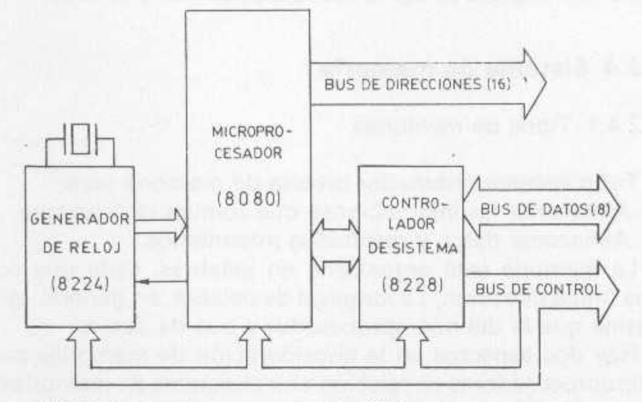


Figura 1.8 Unidad central de proceso de 8080.

1.2.3.3 *Estructura interna*

Según el tipo de microprocesador utilizado, se precisará de más o menos hardware externo para formar una CPU completa. La tendencia de los modernos microprocesadores es la de integrar todas las funciones de la CPU en un solo circuito integrado. Un ejemplo de esta evolución se puede observar en los microprocesadores de Intel 8080 y 8085, tal como se indica en las figuras 1.7 y 1.8; mientras en el sistema 8080 hacen falta tres circuitos integrados para formar la CPU, en el sistema 8085 la generación de reloj y el control de los buses están integrados en la misma pastilla que el resto de la CPU.

La estructura interna de la unidad central de proceso presenta variaciones notables al considerar distintos microprocesadores, por lo que es difícil definir una estructura, no excesivamente simplificada, que se acerque a las distintas alternativas que se han desarrollado.

Además de la unidad de cálculo, sistema decodificador de instrucciones y circuitos temporizadores, se encuentran en la CPU un conjunto de registros internos, en cantidad y con prestaciones variables según la CPU considerada:

a) MAR: Registro de direcciones. Contiene la dirección de la palabra de memoria (o acceso de E/S) seleccionado para una transferencia.
b) PC: Contador de programa. Contiene la dirección de la instrucción en curso.
c) IR: Registro de instrucción. Memoriza el código de la operación de la instrucción que se está ejecutando.
d) MDR: Registro de datos de entrada y salida de la CPU.
e) AC: Acumulador. Es el registro base en operaciones aritméticas y lógicas. En general es también registro fuente y destino de transferencias con memoria y sistema de E/S. Existen máquinas con más de un acumulador.
f) SR: Registro de estado. Está formado por un conjunto de biestables distribuidos por la CPU y que se pueden tratar de una forma conjunta y/o individualmente. Indican condiciones aritméticas (acarreo, desbordamiento, resultado cero, condiciones de comparaciones, paridad...) y de estado (máscaras de interrupción, peticiones pendientes...).
g) SP: Puntero de stack, utilizado para la gestión del stack (pila LIFO) en máquinas que permiten esta estructura.
h) Registros de utilización general. Son registros internos, accesibles por programa, para el almacenamiento temporal, punteros de memoria, registros de índice, etc.
i) Registros auxiliares para uso interno de la CPU (almacenamiento temporal) y no accesibles al programador.

Como ejemplo de estructura interna de un microprocesador concreto, se presenta a continuación el M6800 (Motorola).

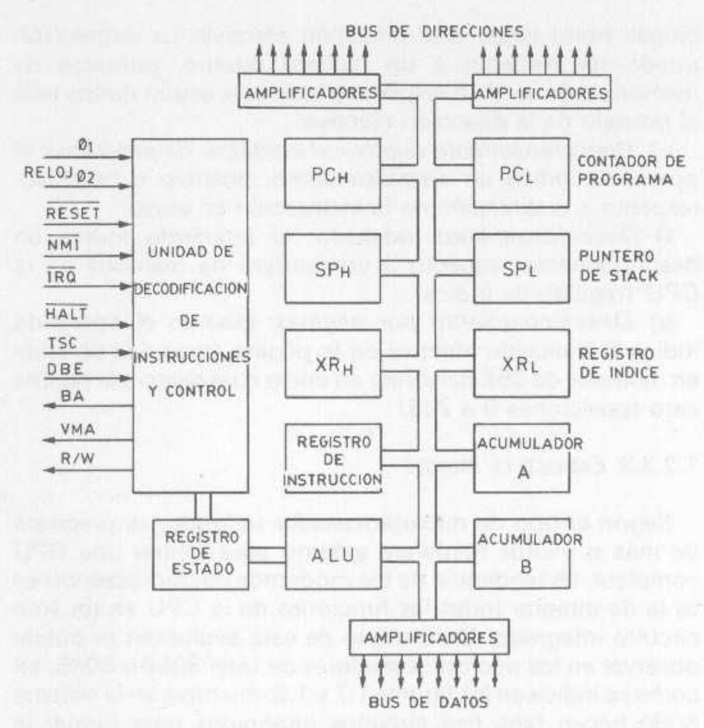


Figura 1.9 Diagrama de bloques de la estructura interna del 6800.

Tal como se observa en la figura 1.9, el 6800 dispone de un único bus interno para establecer la ruta de datos entre los distintos registros de la CPU.

El bus de direcciones está formado por 16 líneas, demultiplexadas, lo cual permite un direccionamiento máximo de 64K palabras (entre registros de memoria y accesos de E/S).

El bus de datos es bidireccional y de una anchura de palabra de 8 bits. Tanto el bus de datos como el bus de direcciones pueden ser llevados al estado de alta impedancia, permitiendo la realización de transferencias por DMA.

El bus de control dispone de diversas líneas para el control del microprocesador y de la memoria y periféricos:

a) Señales de reloj ($\phi 1$, $\phi 2$): reloj de dos fases que debe ser generado por un sistema externo (MC6270/71). La frecuencia máxima es de 1 MHz.
b) Señales para el control de transferencias:
   — Lectura/escritura (R/W). Indica el tipo de transferencia.
   — Validación de direcciones de memoria (VMA). Se utiliza, en combinación con $\phi 2$ para activar los sistemas externos a la CPU.
c) Señales para el control de los buses:
   — DBE: Habilitación del bus de datos. Controla la activación del bus de datos. Normalmente está gobernada por $\phi 2$.
   — TSC: Línea de control que fuerza al bus de direcciones y la línea R/W a estado de alta impedancia.
   — $\overline{\text{Halt}}$: Ocasiona el paro del microprocesador y el paso del bus de direcciones, datos y R/W al estado de alta impedancia.
   — BA: Indica que el bus de direcciones está disponible, debido a la activación de $\overline{\text{Halt}}$ o de la instrucción WAIT.

   Estas señales se utilizan para controlar las transferencias por acceso directo a memoria.
d) Señales que modifican la secuencia de ejecución, interrumpiendo el procesador:
   — Reset: ocasiona la inicialización del sistema.
   — $\overline{\text{NMI}}$: interrupción no enmascarable.
   — $\overline{\text{IRQ}}$: interrupción enmascarable por programa.

El tratamiento subsiguiente es vectorizado de forma indirecta, es decir, el PC se carga con el contenido de unas posiciones fijas de memoria. En el caso de interrupciones, se guarda el estado (registros) en el stack.

El diagrama de flujo de operación interna de la CPU se ilustra en la figura 1.10.

El 6800 dispone de un conjunto de 72 instrucciones y muchas de ellas pueden combinarse con distintos modos de direccionado. Cada ciclo de instrucción se subdivide en varios ciclos de máquina (entre 2 y 12) y cada ciclo en dos subciclos, marcados por los niveles altos de las dos fases $\phi 1$ y $\phi 2$, no solapadas. Durante la fase $\phi 1$ se activa el bus de direcciones, leyéndose la instrucción o el dato cuando está activa, la fase $\phi 2$. La ejecución de las operaciones internas asociadas se realiza en el próximo ciclo de $\phi 1$, lo cual permite un cierto solapamiento entre búsqueda y ejecución de una instrucción.

Para el desarrollo de sistemas basados en el 6800 se dispone de toda una familia de componentes compatibles (memorias ROM, RAM, interfaces, controladores, etc.), y de sistemas de desarrollo (EXORciser) para la puesta a punto de hardware y software, cross-ensambladores, simuladores, etc.

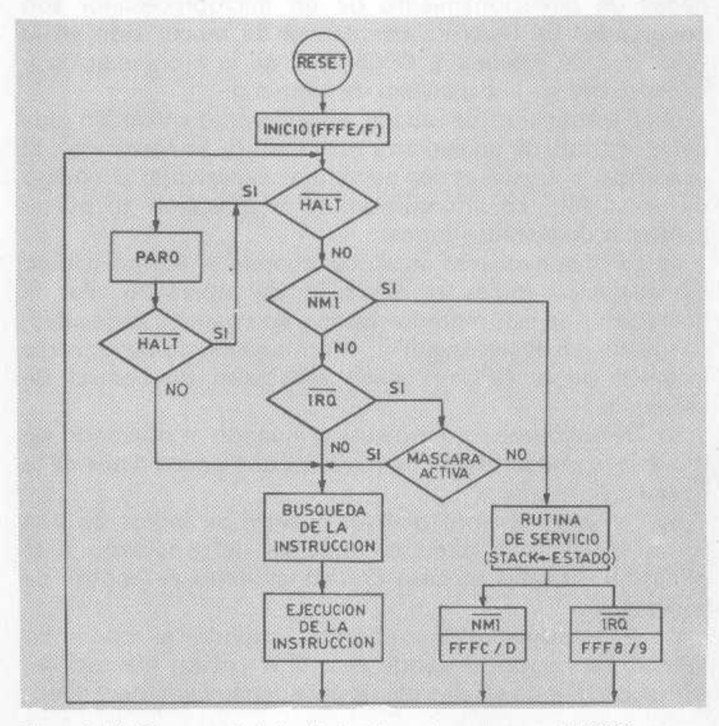


Figura 1.10 Diagrama de flujo del funcionamiento interno del 6800.

1.2.4 Sistema de memoria

1.2.4.1 Tipos de memorias

Todo sistema ordenador precisa de memoria para:
— Almacenar las instrucciones que forman el programa.
— Almacenar datos y resultados intermedios.

La memoria está organizada en palabras, cada una con una única dirección. La longitud de palabra, en general, es la misma que la del microprocesador y bus de datos.

Hay dos aspectos en la consideración de memorias para microprocesadores en relación con el sistema de memoria de miniordenadores:

1) Los sistemas basados en microprocesador deben ser compactos y baratos, por lo que la organización de memoria, en cuanto a eficiencia, cobra una vital importancia.

2) Las aplicaciones de los microprocesadores son, típicamente, de carácter dedicado, es decir, el programa es fijo y está grabado en memorias no volátiles y de acceso sólo para lectura.

Las memorias utilizadas en microcomputadores son, en general, de semiconductores y de acceso aleatorio. Otros tipos de memorias utilizadas para aplicaciones especiales son las memorias de acceso secuencial [registros de desplazamiento, dispositivos de acoplo de carga (CCD) y memorias de burbuja magnética], caracterizadas por tener una gran capacidad de almacenamiento y un tiempo de acceso relativamente elevado. Finalmente existen las memorias auxiliares de gran capacidad y bajo coste, como cintas magnéticas, discos y cassetes, que por su forma de acceso son consideradas como periféricos del microcomputador.

Las memorias de semiconductores de acceso aleatorio se fabrican utilizando diversas tecnologías (TTL, STTL, PMOS, NMOS, CMOS, I^2L, ECL...) con características distintas en tiempos de acceso, posibilidades de integración y consumo.

Las memorias de solo lectura (ROM) se utilizan en microcomputadores para almacenar programas, tablas y constantes.

Existen tres tipos básicos de memorias ROM:

1) *ROM de máscara:* son memorias programadas en el proceso de fabricación e inalterables posteriormente. Son las más económicas en coste por bit pero sólo para grandes cantidades, ya que el coste fijo es elevado, por lo que son utilizadas en la producción de grandes series.

2) *ROM programable* (PROM): son memorias, que pueden ser programadas por el usuario pero no modificadas, ya que el proceso de programación es irreversible. Son más aptas que las memorias ROM de máscara para pequeñas series.

3) *ROM reprogramables* (EPROM): son memorias que pueden ser grabadas y borradas o modificadas posteriormente. Las memorias EPROM son especialmente aptas para la realización de prototipos y pequeñas series.

Las memorias ROM existen con capacidades muy diversas, observándose un rápido crecimiento en los niveles de integración, llegándose hasta el momento a capacidades de 64K bits.

Las memorias de escritura y lectura (RWM o RAM) tienen como finalidad el almacenamiento de datos y resultados intermedios o también se utilizan como memoria de programa para zonas del mismo que deben ser modificadas en curso de ejecución, para la fase de puesta a punto de un sistema, o para sistemas en los que el programa es cargado al inicializar el sistema a partir de un medio auxiliar de memoria.

Las memorias RAM son volátiles, es decir, la información almacenada se pierde si falla la alimentación del sistema. Si es preciso salvar la información en caso de fallo de tensión se deberá disponer de un sistema de baterías para alimentación de la RAM. En este caso las RAM de tecnología CMOS son especialmente útiles por su bajo consumo.

Existen dos tipos de memoria RAM de acceso aleatorio:

1) *RAM estáticas:* la información se almacena en un biestable, de forma que el contenido se mantiene mientras haya alimentación.

2) *RAM dinámicas:* la información se mantiene en forma de carga en la capacidad puerta-sustrato de un transistor MOS. Esta carga se disipa con el tiempo (del orden de milisegundos) por lo que es necesario un refresco periódico consistente en un ciclo de lectura/escritura en cada bit de memoria, realizado de forma paralela con todos los bits de una misma columna a la vez.

La principal desventaja de las memorias dinámicas es la necesidad de disponer de un circuito de refresco externo, en general complejo. Esta desventaja va desapareciendo al irse desarrollando circuitos integrados de memoria dinámica con circuitos de control de refresco incorporado. Además, algunos microprocesadores (Zilog Z-80) incorporan en sus salidas de control señales para facilitar la circuitería de control de memorias dinámicas.

Las ventajas de la memoria dinámica es que permiten una mayor densidad de integración (actualmente del orden de 64K bits frente a 16K en las estáticas), al precisar menos transistores por célula elemental de memoria, por lo que son más baratas que las RAM estáticas de la misma capacidad y consumen menos potencia.

Las memorias dinámicas acostumbran a organizarse en módulos de Nx1 bits, lo cual implica que se debe utilizar un mínimo de 8 circuitos integrados para disponer de una memoria RAM para un microprocesador de 8 bits de longitud de palabra. Esto es un inconveniente para sistemas reducidos pero no lo es para sistemas de gran capacidad de memoria, ya que la organización de Nx1 es la más económica, para la misma capacidad.

1.2.4.2 *Interconexión del sistema de memoria*

Existen dos tipos de estructura conceptualmente distintos en cuanto a la forma de considerar la memoria en un microcomputador:

1) *Estructura Harvard:* cuando la memoria de programa (pasiva) recibe un tratamiento distinto de la memoria de datos (activa). Las diferencias pueden consistir en el modo de direccionado de forma que las líneas de control (o algunas de ellas) que activan cada subsistema (ROM, RAM) no son comunes, pudiéndose llegar a una total diferenciación de los buses de comunicación, lo cual permitiría un cierto solapamiento entre los ciclos de búsqueda y ejecución de las instrucciones.

Algunos ejemplos de este tipo de estructura son: Rockwell PPS-8, Intel 8048, Scientific Microsystems SMS300...

2) *Estructura Princeton o Von Newman:* cuando el tratamiento de memoria es único, sin distinción entre datos e instrucciones. Este sistema es el más sencillo y el más flexible, y es el que emplean la mayoría de los microcomputadores.

Al interconectar la CPU con el sistema de memoria se deberán considerar tres tipos de información:

1) *Direcciones:* sirven para seleccionar una posición de memoria determinada, y deberán mantenerse estables durante todo el ciclo de memoria (en lectura o escritura).

2) *Datos:* son las líneas que forman el bus de datos y a través de ellas se realiza la transferencia de información.

3) *Control:* son las señales que activan y sincronizan la transferencia. Las señales de control a considerar, en general, son:

a) Líneas de selección: se obtienen a partir de la decodificación de las líneas de direcciones más significativas, no utilizadas para el direccionamiento interno de cada circuito integrado. La selección se puede dar a dos niveles:
   — Selección de módulos de memoria.
   — Selección de circuitos de memoria dentro de cada módulo.

b) Líneas de validación de transferencia: son señales de activación de la lectura o escritura. Indican cuando el

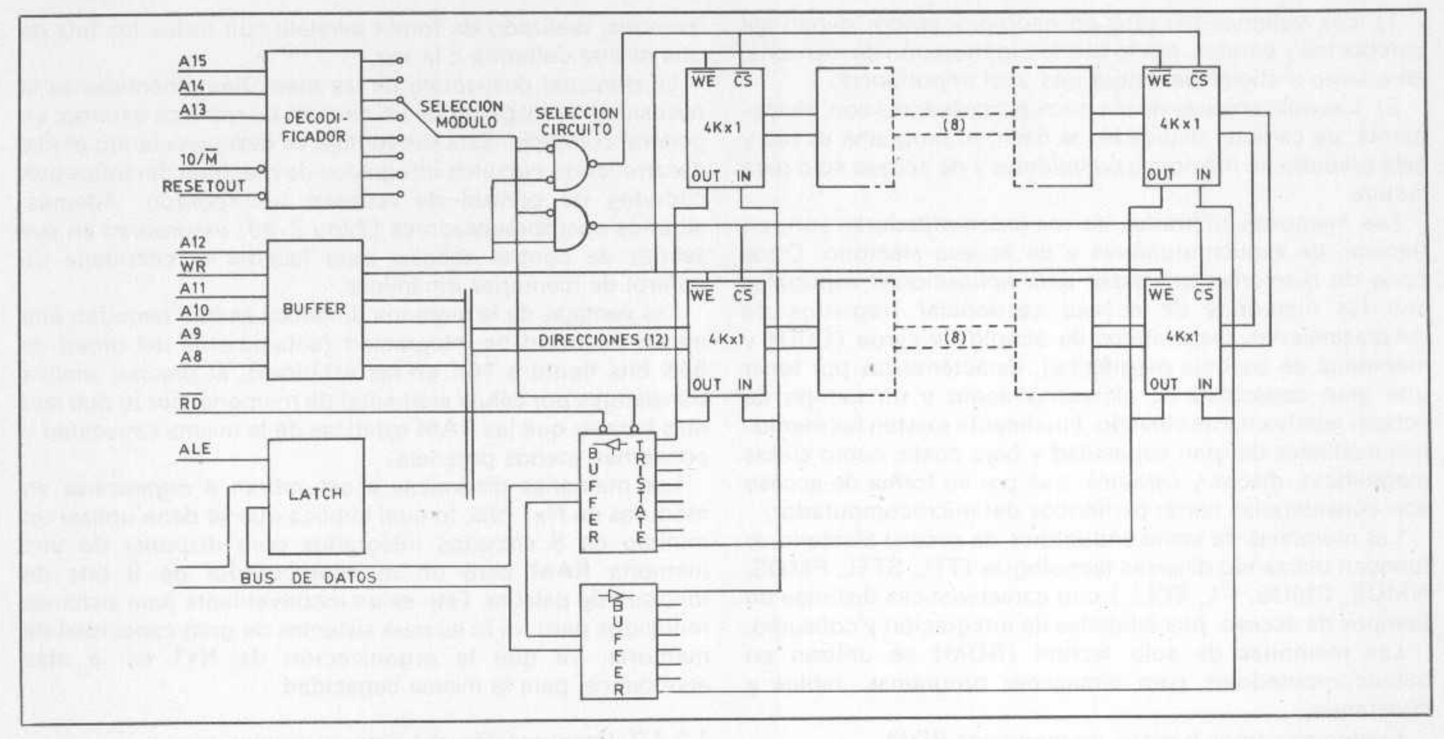


Figura 1.11 Interfaz de un módulo de RAM de 8K byte al bus de 8085.

microprocesador entra los datos (lectura) o que los datos enviados por la CPU son válidos (escritura).

c) Líneas de sincronización: algunos microprocesadores permiten alargar el ciclo de lectura/escritura a petición de la memoria seleccionada, pudiéndose de esta forma acomodar el tiempo de acceso al ciclo de máquina.

En la figura 1.11 se muestra un diagrama de bloques de la conexión de un subsistema de memoria RAM de 8K bytes a un sistema microcomputador. Las unidades de memoria se suponen de 4Kx1 bits con entrada y salida de datos separadas.

1.3 SISTEMA DE ENTRADA/SALIDA

1.3.1 Introducción

El sistema de entrada y salida permite la comunicación del microcomputador con el mundo exterior.

Debido al tipo de aplicaciones, en general, en las que son utilizados los microprocesadores, la capacidad y eficiencia en el tratamiento del sistema de E/S es una de las características principales de un sistema microcomputador.

Se denomina interfaz al sistema hardware-software que permite la comunicación con un periférico determinado, es decir, el conjunto de circuitos (hardware) y programas (software) que se utilizan para establecer la comunicación.

La forma concreta de realizar una interfaz dependerá de las alternativas que se consideren. En principio, dentro del balance «hard-soft» se deberá potenciar el software ya que, en general, el incremento en coste de memoria es inferior al incremento del hardware preciso para realizar la misma función, además de la mayor flexibilidad que permite el software. En otros casos esta afirmación no estará justificada ya que el coste de programación puede ser decisivo. Por otra parte, las tareas de entrada y salida pueden consumir excesivo tiempo de máquina, por lo que puede ser precisa la utilización de circuitería externa, compleja y especializada, a fin de posibilitar la ejecución de estas tareas u otras de proceso que no se podrían ejecutar si el microprocesador tuviese que hacerse cargo de toda la gestión y control de las operaciones de transferencia.

Existen dos tipos de información en la comunicación microprocesador-periférico:

1) *Datos:* entrada de información para proceso y salida de resultados.

2) *Control:* salida de señales para el gobierno de los periféricos y entrada de información del estado de los mismos.

La flexibilidad en el tratamiento de esta información dependerá de varios factores:

a) Instrucciones del microprocesador aplicables al sistema de E/S.

b) Posibilidad de delegación de las tareas de control de la transferencia a circuitos de adaptación externos al microprocesador y controladores de periféricos.

c) Técnicas de transferencia utilizadas.

Las funciones que deberá realizar el sistema de interfaz o adaptación son:

— Identificación de direcciones, a fin de establecer la conexión con el bus de datos y control cuando un acceso concreto de E/S es seleccionado.

— Interpretación de órdenes. En general, las órdenes enviadas directamente al sistema de E/S por el microprocesador se reducen a señales de escritura y lectura, ya decodificadas o que precisan un pretratamiento sencillo. En otros casos, la estructura de E/S es más compleja y se envían a los periféricos señales de control a partir de las cuales se debe interpretar la operación a realizar (lectura, escritura, borrado, test, activación...).

— Adaptación física entre los dos sistemas, microcomputador y periférico. Esta función es realizada por los transmisores y receptores de línea, convertidores tensión-corriente y viceversa, optoaisladores, etc. Si la transmisión de los datos se realiza en serie, será preciso considerar, además, la conversión de formato de la información en ambos sentidos.

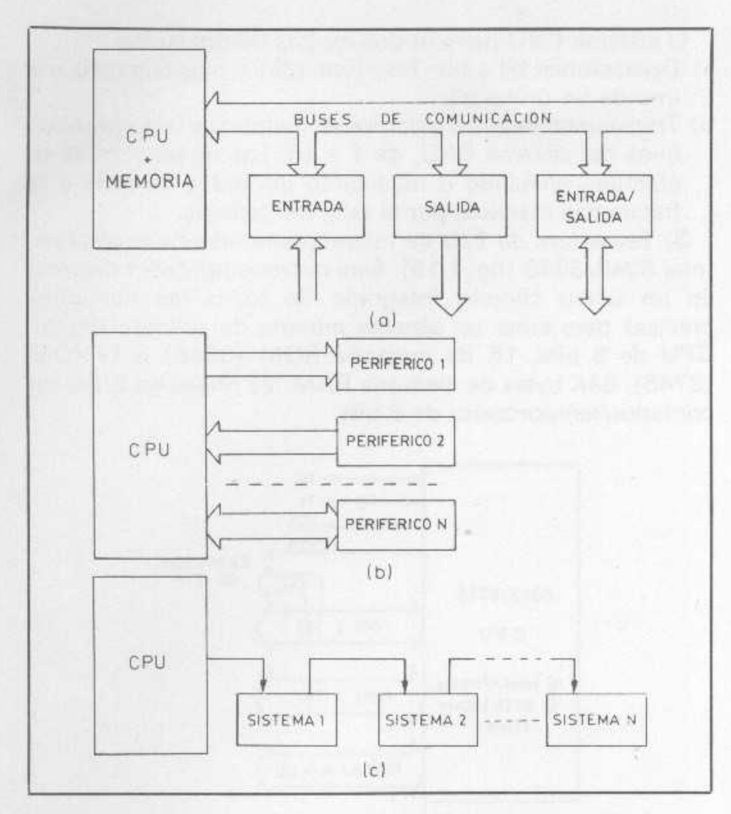


Figura 1.12 Estructuras de E/S: a) línea compartida; b) estrella; c) cadena.

— Temporización de la transferencia a fin de controlar el flujo de información entre sistemas de forma ordenada y eficaz. En esta función intervienen los protocolos de comunicación y las técnicas de transferencia.

1.3.2 Buses de entrada/salida

Los distintos periféricos estarán enlazados con la CPU siguiendo una de las estructuras siguientes:

1) *Línea compartida (party-line):* Los diversos módulos de entrada y salida comparten en el tiempo unas líneas únicas, tal como se indica en la figura 1.12*a*. En general, estas líneas son un subconjunto de los buses externos, por lo que el sistema periférico comparte el bus de direcciones, datos y control con el sistema de memoria. Esta estructura es la más utilizada, ya que reduce significativamente el número de líneas que precisa la CPU para la gestión de E/S y posibilita a la vez una concepción modular del sistema.

2) *Estructura radial (star):* En este caso cada periférico se conecta a la CPU a través de líneas dedicadas, tal como se ilustra en la figura 1.12*b*. Este sistema está muy limitado por la cantidad de terminales que se precisan en la CPU. Se utiliza en sistemas en los que la CPU posee algunas líneas de entrada/salida y en el caso de microcomputadores integrados en una sola pastilla, donde la incorporación de memoria ROM y RAM en el mismo circuito integrado que el resto de la CPU, permite disponer de terminales para funciones de E/S.

3) *Estructura de cadena (daisy-chain):* En este caso las señales se propagan de un periférico a otro, tal como se muestra en la figura 1.12*c*. Cuando un periférico concreto es seleccionado, bloquea la transmisión de las señales al resto del sistema. Este método se utiliza en algunos microcomputadores (Fairchild F-8, Rockwell PPS-8) para disponer de un sistema jerárquico de interrupciones, estableciéndose los niveles según la posición en la cadena.

1.3.3 Software de entrada y salida

La gestión de entrada y salida, en su aspecto de programación puede considerarse a dos niveles:

1) *Instrucciones de entrada y salida.* Sirven para realizar la transferencia de información bajo control de programa.

2) *Rutinas de entrada y salida.* Son tareas cuyas funciones están íntimamente ligadas al sistema de E/S, y entre las que se podrían destacar: temporizaciones por programa, rutinas relacionadas con el servicio de interrupciones, inicialización de transferencias por acceso directo a memoria, gestión de tampones (buffers) de memoria para captura de datos, rutinas de empaquetamiento y desempaquetamiento de datos, rutinas de detección de errores, cambios de formato, etc.

En la mayoría de los microcomputadores no existen instrucciones específicas de E/S, ya que no se diferencia entre memoria y registros de E/S, por lo que las mismas instrucciones son aplicables a ambos sistemas. En otros microcomputadores existen instrucciones que hacen referencia exclusiva al sistema de E/S y pueden ser de los tipos siguientes:

a) Transferencia de datos en paralelo: El código de operación indica si la operación es de entrada o de salida y, en general, indica de forma implícita el registro de la CPU (el acumulador en la mayoría de los casos) que interviene en la transferencia.

En máquinas con poca capacidad de direccionado, en cuanto a accesos de E/S, la misma palabra del código de operación puede indicar la dirección del registro periférico seleccionado. En máquinas de mayor capacidad de direccionamiento, o bien, el código de operación indica un registro de la CPU que sirve de base para un direccionamiento indirecto, o la dirección del periférico se da de forma directa en palabras sucesivas al código de operación.

b) Transferencia, seguida de un test de una línea externa y salto (skip) condicional según el resultado del test. Este tipo de instrucciones facilita el diálogo con el periférico en transferencias programadas (ej.: Mostek 5065).

c) Modificación. Se ofrece la posibilidad de modificar algunos bits de un registro de salida, sin modificar el resto, con una sola instrucción. Consiste en realizar operaciones lógicas (AND para la puesta a cero de un bit y OR para la puesta a uno) entre vías de acceso (ports) y datos inmediatos (ej.: Intel 8048).

d) Envío de órdenes especiales a periféricos. En este caso el código de operación contiene información que se envía al periférico a fin de desencadenar una operación específica, como puesta a uno, puesta a cero, complemento de biestables de control, etc.

e) Instrucciones de entrada y salida bit a bit. En máquinas con esta estructura de E/S (9900), se posibilita el direccionamiento y acceso a un bit de entrada o de salida, además del acceso a una palabra completa.

1.3.4 Estructura de entrada y salida

En muchos microcomputadores se encuentra una estructura de buses únicos, de forma que no se diferencia entre memoria y E/S (Memory-mapped I/O). En estos casos, en cuanto a tratamiento e instrucciones aplicables, son equivalentes, para la CPU, una palabra de memoria o un registro de E/S.

En otros microprocesadores, además de posibilitar el tratamiento anterior, se dispone de una estructura de E/S especializada, disponiéndose de líneas de control dedicadas

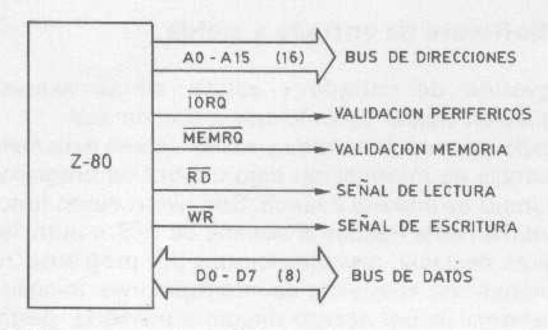


Figura 1.13 Microprocesador Z-80: Líneas de interfaz para transferencias en paralelo.

a la gestión de transferencias con el sistema de E/S e incluso de líneas particulares de datos y/o direcciones.

Citando algunos ejemplos de estructuras particulares de E/S, sin hacer referencia a sistemas de interrupción y DMA que se tratarán más adelante, cabría citar, entre otros:

1) Estructura de transferencias en paralelo con direccionamiento independiente del subsistema de memoria: Zilog Z-80 (fig. 1.13). El Z-80 permite una estructura de vías de acceso (ports) de E/S que no interfiere en el campo de direccionamiento de memoria (I/O mapped). El acceso al sistema de E/S se realiza a partir de instrucciones específicas las cuales generan ciclos de lectura o escritura que se diferencian de los ciclos de acceso a memoria en que se activa la señal $\overline{IORQ}$ en lugar de $\overline{MEMRQ}$ y, además, se inserta automáticamente un estado de espera.

El Z-80 dispone de un repertorio de 158 instrucciones incluyendo las 78 del 8080. Las aplicables específicamente a E/S permiten la entrada y salida de datos con direccionamiento directo e indirecto en forma de transferencias únicas (palabras) o repetitivas (bloques).

2) Direccionamiento de líneas de E/S bit a bit: TMS 9900 (fig. 1.14). El 9900 dispone de una unidad de registro de comunicaciones (CRU) que permite direccionar directamente 4096 bits de entrada y otros tantos de salida.

Las líneas dedicadas al interfaz con el sistema CRU son:

— Direcciones: 12 bits (A3-A14) compartidos con el sistema de memoria.
— Datos: CRUIN (entrada) y CRUOUT (salida).
— Control: CRUCLK, pulso que valida las operaciones de salida.

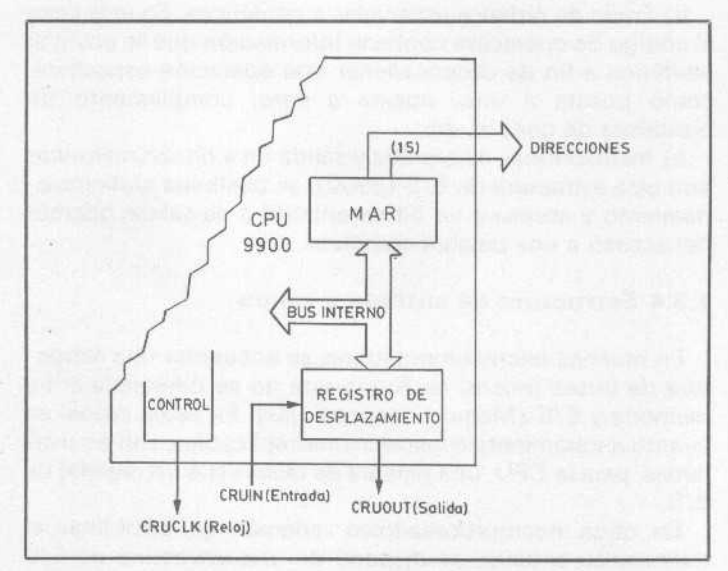


Figura 1.14 Estructura de E/S bit a bit del 9900.

El sistema CRU permite dos modos de operación:

a) Operaciones bit a bit: Test (entrada) y puesta a cero o a uno de un único bit.
b) Transferencias entre palabras de memoria y bits consecutivos del sistema CRU, de 1 a 16. Las transferencias se efectúan enviando o recibiendo los datos en serie a la frecuencia marcada por el reloj del sistema.

3) Estructura de E/S de microcomputadores integrados: Intel 8048/8748 (fig. 1.15). Este microcomputador dispone en un único circuito integrado de todas las funciones precisas para crear un sistema mínimo de proceso digital: CPU de 8 bits, 1K de memoria ROM (8048) o EPROM (8748), 64K bytes de memoria RAM, 27 líneas de E/S y un contador/temporizador de 8 bits.

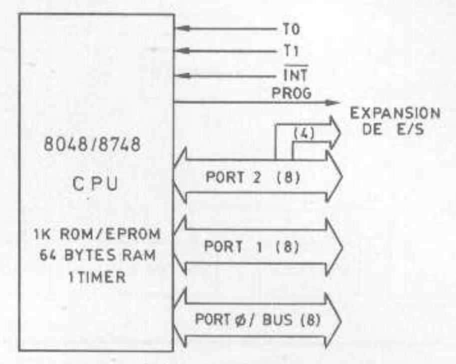


Figura 1.15 Líneas de E/S del 8048/8748.

El port 0 puede configurarse como acceso de entrada o salida, o constituirse en bus de datos y direcciones para ampliaciones de ROM, RAM o E/S.

Los ports 1 y 2 pueden configurarse como entradas o salidas de una forma independiente bit a bit.

Las instrucciones con referencia a ports de E/S pueden ser de transferencia en paralelo (IN, OUT) o de modificación (AND, OR) lo cual permite el tratamiento de E/S a nivel de palabra y de bit. El port 2 (líneas 0-3) puede ser considerado, además, como los 4 bits más significativos del bus de direcciones para ampliaciones de memoria ROM, a la vez que permite la interconexión con circuitos expansores de E/S (8243).

Las tres líneas de entrada adicionales (T0, T1 e $\overline{INT}$) son consultables por programa y pueden configurarse como salida de reloj del sistema (T0), entrada al contador interno (T1) y petición de interrupción enmascarable ($\overline{INT}$).

1.3.5 Interfaces programables

1.3.5.1 *Características*

Varias familias de microcomputadores disponen de circuitos integrados LSI compatibles con el resto de los elementos del sistema a fin de potenciar el sistema de entrada/salida.

Las características fundamentales de estos subsistemas son:

1) *Facilidad de interconexión a los buses de comunicación:* Por ser elementos compatibles con el resto del sistema en cuanto a señales y temporizaciones, es muy fácil conectarlos a los buses, requiriendo en general solamente un sistema de decodificación de direcciones. En la figura 1.16 se muestra la conexión del circuito de Intel PPI 8255 al bus del sistema MCS 8080.

2) *Posibilidad de programación:* De esta forma se logra disponer de un cierto número de opciones de funcionamien-

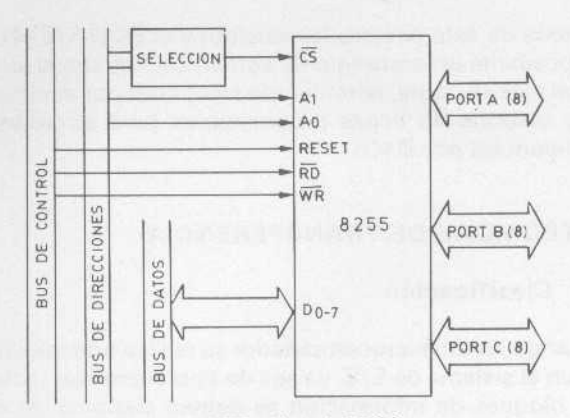


Figura 1.16 Interfaz del PPI 8255 al bus del 8080.

to. En general, la programación consiste en cargar de una forma determinada un registro interno de modo de operación, lo cual da, además, la posibilidad de una reprogramación dinámica, es decir, en fase de operación. Esta posibilidad no siempre es utilizada, pero una programación inicial que permita adaptar el sistema a las necesidades particulares de cada aplicación amplía los campos de utilización de un determinado circuito, dando como resultado una modularidad y reducción en coste en comparación con el desarrollo de circuitos especiales para cada configuración.

3) *Inteligencia distribuida:* La posibilidad de que estos circuitos posean una cierta capacidad de decisión para desarrollar funciones de control permite descargar al microprocesador de estas tareas, simplificando de esta forma el programa y/o posibilitando determinadas prestaciones del sistema que serían imposibles de conseguir si la unidad central tuviera que hacerse cargo de toda la gestión del sistema de E/S.

1.3.5.2 *Clasificación*

Por la función que realizan, se pueden dividir en tres categorías:

1) *Interfaces dedicadas:* son circuitos especializados para una función específica, aunque en general permiten un cierto número de opciones. En esta categoría se incluyen temporizadores, controladores de sistema (interrupciones y acceso directo a memoria) y controladores de periféricos (disco flexible, pantalla TRC, teclados, protocolos, etc.).

2) *Interfaces generales:* son circuitos de tipo más universal que los anteriores y tienen como misión posibilitar la transferencia y diálogo entre el microprocesador y los periféricos.

Funcionalmente existen dos tipos distintos:

a) Interfaces para transmisión en paralelo: permiten disponer de un número elevado de líneas de entrada y salida que permitirán el intercambio de información así como el envío de órdenes y recepción de estado de una forma sencilla, flexible y rápida.

Algunos ejemplos de este tipo de interfaces son: Intel PPI 8255, Motorola PIA 6820, Texas Instruments TMS 9901, Rockwell PDC 10453...

b) Interfaces para transmisión en serie: permiten utilizar canales de transferencia serie de un modo transparente al microprocesador, es decir, se descarga a la unidad central de las tareas de serialización y sincronización. Pueden ser síncronas y/o asíncronas según permitan un modo u otro de transmisión.

Algunos ejemplos de este tipo de interfaces son: Intel 8251, Motorola ACIA 6850 y SSDA 6852, Texas Instruments TMS 9902 y TMS 9903, Rockwell SDC 10930...

3) *Microcomputadores especializados de entrada y salida:* en principio, cualquier microprocesador puede considerarse a modo de "esclavo" de otro, realizando tareas bajo control del "director". Estas tareas pueden ser cualesquiera y por tanto la gestión de entradas y salidas es una de ellas. Un microcomputador especialmente diseñado con esta finalidad es el Intel 8041/8741. El grado de complejidad de las tareas que puede realizar este sistema es muy elevado, por lo que no es propiamente una interfaz programable, sino más bien un procesador dedicado.

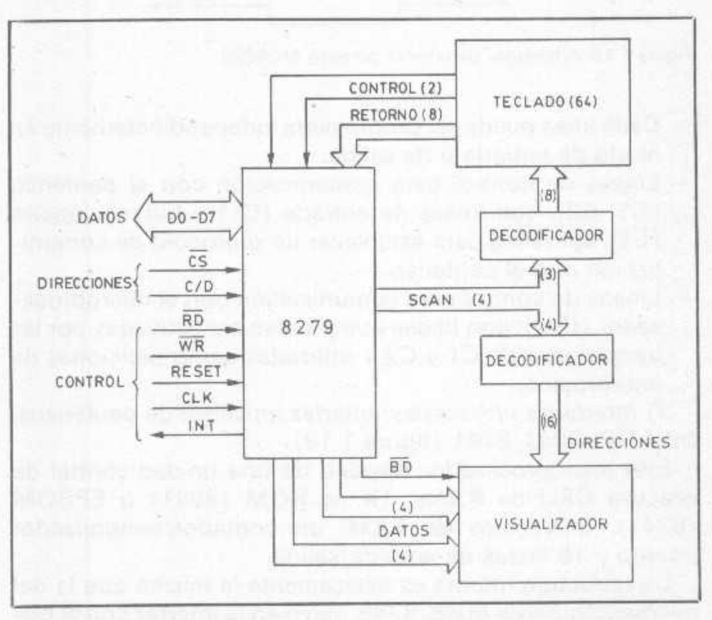


Figura 1.17 Interfaz de un teclado y visualizador con un controlador especializado.

1.3.5.3 *Ejemplos*

Aunque estos tipos de interfaces programables, especializadas o no, serán ampliamente descritos en otros capítulos, se presentan a continuación algunos ejemplos de circuitos periféricos LSI encontrados en el mercado:

1) *Interfaces especializadas:* controlador de visualizador y teclado Intel 8279 (figura 1.17).

Este sistema permite la gestión descentralizada de un teclado formado por una matriz de hasta 64 teclas (ampliable a 128) y un sistema de visualización de 16×8 segmentos (visualizadores alfanuméricos o LED).

La gestión del teclado se efectúa por muestreo periódico (scanning), siguiendo una de varias opciones programables. Las entradas son filtradas por el propio controlador. Posee una memoria interna de teclas pulsadas, así como un registro de estado que pueden ser consultados por el microprocesador.

El refresco del sistema multiplexado de visualización se hace de una forma automática a partir de la información guardada en la RAM interna con lo que se releva al microcomputador de esta tarea, comportándose el sistema de visualización como si fuese estático.

2) *Interfaces generales:* Adaptador de interfaz paralelo, Motorola PIA MC-6820 (figura 1.18).

Cada uno de los canales dispone de las siguientes líneas:

— Líneas de datos (P0-P7): son las líneas utilizadas para el intercambio de información (datos) con el periférico.

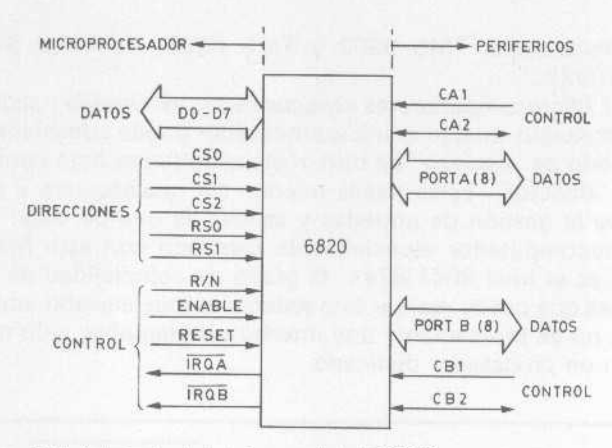


Figura 1.18 Adaptador de interfaz paralelo MC6820.

Cada línea puede ser programada independientemente en modo de entrada o de salida.

— Líneas de control para comunicación con el periférico (C1, C2): son líneas de entrada (C1) y bidireccionales (C2) utilizadas para establecer un protocolo de comunicación con el periférico.

— Líneas de control para comunicación con el microprocesador (IRQ): son líneas que pueden ser activadas por las transiciones de C1 y C2 y utilizadas como peticiones de interrupción.

3) *Interfaces universales:* Interfaz universal de periféricos, Intel UPI 8041/8741 (figura 1.19).

Este microprocesador dispone de una unidad central de proceso CPU de 8 bits, 1K de ROM (8041) o EPROM (8741), 64K bytes de RAM, un contador/temporizador interno y 18 líneas de entrada/salida.

La estructura interna es básicamente la misma que la del microcomputador 8048/8748, salvo en la interfaz con el bus de datos, que está optimizada para la recepción de datos y órdenes y envío de datos e información de estado de una forma asíncrona.

Para la comunicación con periféricos, el sistema dispone de dos vías de acceso (P10-17/P20-27) de 8 bits cuasibidireccionales y que permiten la manipulación en paralelo o bit a bit. Las líneas T0 y T1 son consultables por programa y, además, T1 puede ser programada como entrada del temporizador interno. Esta configuración de E/S es ampliable utilizando los expansores de E/S (8243). Una versión mejorada de este circuito la constituye el 8041A/8741A, el cual posibilita un protocolo de comunicación con el sistema central más eficiente, permitiendo su control por interrupciones y dispone de líneas programables para el control de transferencias por DMA.

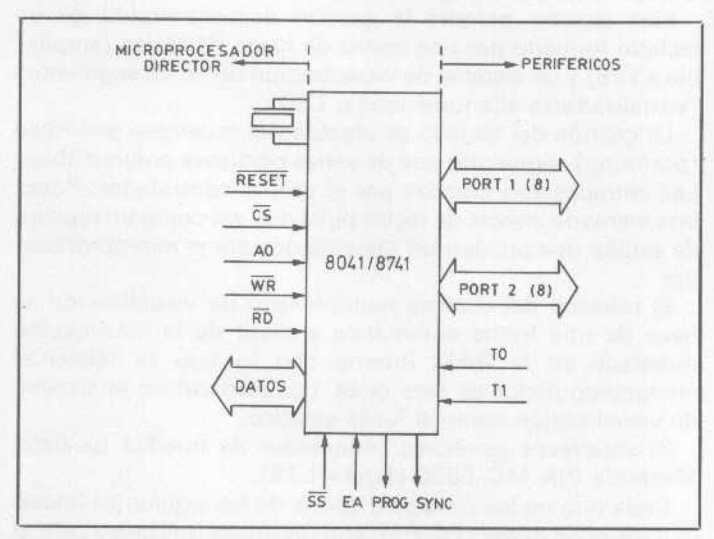


Figura 1.19 Interfaz universal de periféricos UPI 8041/8741.

1.4 TÉCNICAS DE TRANSFERENCIA

1.4.1 Clasificación

Cuando en un microcomputador se realiza una transferencia con el sistema de E/S, ya sea de tipo elemental (palabra) o de bloques de información se deberá disponer de algún sistema que sea capaz de iniciar, realizar y controlar el flujo de información.

Respecto a la forma de ejecución de la transferencia cabe distinguir entre dos alternativas diferenciadas:

1) *Transferencia por programa:* cuando el intercambio de información se realiza vía registros de la CPU por medio de instrucciones de transferencia de datos con el sistema de E/S.

2) *Transferencia por acceso directo a memoria:* cuando se inhibe a la CPU de la tarea de control, realizándose la transferencia bajo control de un sistema externo (controlador de DMA) de forma que la comunicación se establece directamente entre memoria y periférico.

En caso de ejecutarse la transferencia por programa, se pueden considerar dos posibilidades en cuanto a la forma de control o estrategia utilizada:

1) Transferencia síncrona, cuando se realiza por iniciativa del programa y sin consultar al periférico sobre su estado. Este método es el más sencillo en cuanto a necesidades de software y hardware (fig. 1.20) precisos, pero su utilización estará restringida a la gestión de aquellos periféricos que siempre estén dispuestos a aceptar una nueva información (ej.: un visualizador alfanumérico) o a entregar los datos que se le piden (ej.: un pulsador).

2) Transferencia asíncrona, cuando es preciso establecer un diálogo (handshaking) entre CPU y periférico a fin de controlar la ejecución de la transferencia en el instante y de la forma apropiadas. En este caso cabe considerar dos alternativas en cuanto a la forma de control:

a) Por consulta de estado: se pregunta al periférico sobre su disposición para aceptar la transferencia o su necesidad de servicio.

b) Por interrupción: se delega en el periférico la facultad de avisar cuando está dispuesto o si precisa servicio.

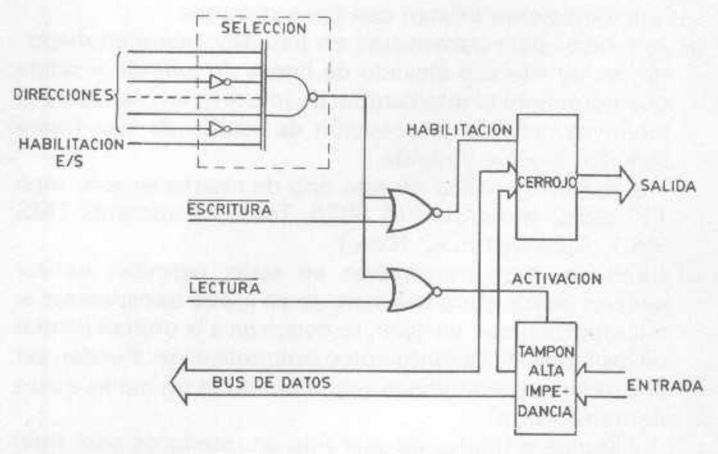


Figura 1.20 Interfaz para una transferencia síncrona en paralelo.

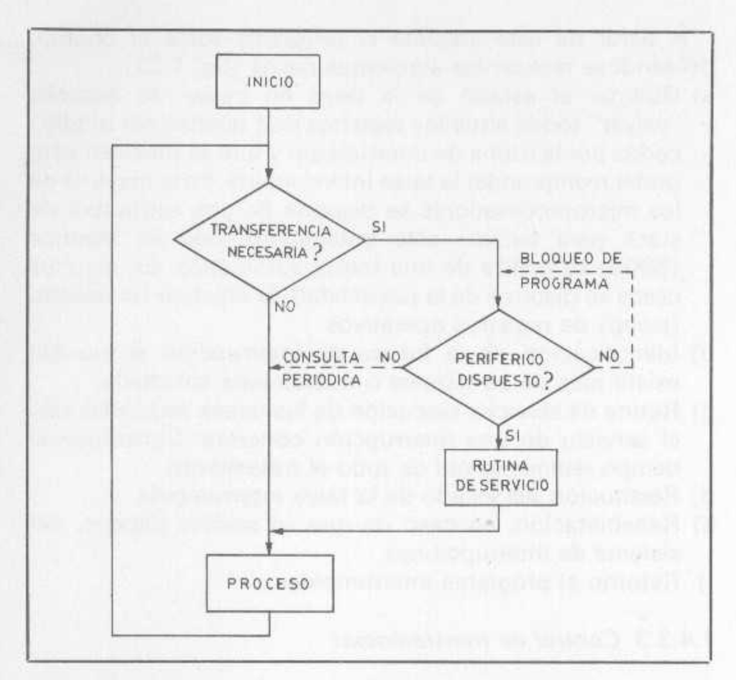


Figura 1.21 Diagrama de flujo del control de transferencias por consulta de estado.

1.4.2 Transferencia asíncrona

En este método de control de transferencias se deberá establecer un mecanismo de sincronismo del trasvase de información entre microcomputador y periférico. En muchos casos esta necesidad viene motivada por las diferencias relativas, en cuanto a velocidad, entre procesador (rápido) y periférico (lento).

Para posibilitar este tipo de diálogo, será preciso que el periférico disponga de un registro de estado que suministre la información adecuada (libre, dato válido...) a requerimiento del microprocesador.

En el caso de control por consulta de estado cabe considerar dos formas de establecerla (fig. 1.21):

1) *Método de bloqueo de programa:* el procesador queda en bucle de espera, consultando continuamente el estado del periférico, en caso de que se precise efectuar una transferencia. Con este método, y según sea el período de respuesta del periférico, se puede perder mucho tiempo consultando el estado, con lo que puede quedar considerablemente disminuida la potencia del sistema. En aplicaciones en tiempo real, este sistema, sin modificaciones, tiene una aplicación muy restringida.

2) *Método de consulta periódica:* se interroga periódicamente la necesidad y posibilidad de efectuar una transferencia. De esta forma no se penaliza el tiempo de proceso, pero no se soluciona el problema de que, debido a que la consulta se realiza por iniciativa del programa, puede ocurrir que el tiempo existente entre el instante en que se precisa la transferencia y la ejecución de la misma sea superior al intervalo dentro del cual el periférico necesita el servicio.

A fin de efectuar de una forma efectiva el diálogo entre emisor y receptor (microcomputador o periférico en cada caso) se dispondrá de un hardware adicional consistente en líneas especiales de control (handshaking) independientes de las líneas de datos, que indiquen al receptor cuando el emisor ha enviado una información y ésta es válida y al emisor cuando el receptor la ha recibido y está dispuesto para una nueva transferencia.

Como ejemplo se podría considerar el protocolo de comunicación que seguirían los dos sistemas representados en la figura 1.22:

a) Entrada: El periférico activa la señal DAV (dato válido) con lo que la interfaz carga el registro de entrada y responde desactivando la línea DAC (dato aceptado), indicando al periférico que no puede enviar otra información. Se modifica el registro de estado de forma que cuando es consultado por el microprocesador, éste se entera de que existe una información para ser tratada. Cuando el microprocesador lee el registro de datos, se activa DAC, a lo que el periférico contesta desactivando DAV, quedando el sistema dispuesto para una nueva transferencia.

b) Salida: En este caso, la señal DAV se activa cuando el microprocesador escribe en el registro de salida. Cuando el periférico contesta mediante la activación de DAC, se modifica el registro de estado, de forma que el microprocesador pueda enterarse por consulta que ya se puede enviar otra información. La desactivación de DAV por parte del microprocesador y la consiguiente de DAC por parte del periférico deja al sistema en el estado inicial.

En ambos casos puede ser programada la interfaz de forma que se active la línea de INTR además de los bits de estado, con lo que se evitaría al microprocesador la tarea de consulta y se controlaría la transferencia por interrupciones. Las técnicas de control por interrupción se presentarán en el siguiente apartado.

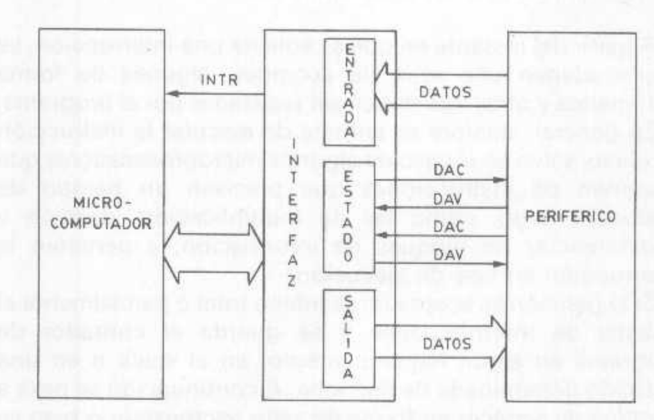


Figura 1.22 Hardware de control asíncrono de transferencias.

1.4.3 Sistema de interrupciones

1.4.3.1 *Introducción*

Una interrupción es una primitiva de control externa de tipo hardware que permite "avisar" al microprocesador de la ocurrencia de un suceso que precisa su atención y tratamiento.

Las causas que pueden generar una interrupción son diversas y no siempre están relacionadas con el sistema de entrada y salida. Así, se podrían considerar:

— Interrupciones por errores o averías de máquinas: fallos de alimentación, errores de paridad en memoria, fin de cinta en una perforadora, etc.
— Interrupciones de reloj en sistemas operativos en tiempo real.
— Interrupciones externas de aviso de modificaciones importantes del mundo exterior.
— Interrupciones por causa de programa: recepción de códigos de operación inexistentes, direccionamientos

incorrectos, operaciones imposibles, desbordamiento, intento de escritura en zonas protegidas, etc.

— Interrupciones del sistema de entrada/salida: avisos de fin de entrada de datos en transferencias por DMA, condiciones de periféricos libres o dispuestos, fin de temporizaciones, etc.

En muchos casos existirán zonas de programa en las cuales no se podrá aceptar ninguna interrupción, o solamente algunas de ellas, ya sea por temporizaciones críticas, cuestiones de prioridades u otros motivos.

Para posibilitar esta opción, la mayoría de los microprocesadores disponen de instrucciones que permiten inhibir total o parcialmente el sistema de interrupciones. En este caso, en algunos microprocesadores se recordará la petición a fin de servirla cuando se pueda; en otros deberá ser el propio periférico quien "recuerde" la petición, manteniendo activa la señal hasta que sea servida.

En el control y gestión del sistema de interrupciones se deberán considerar diversos aspectos:

a) Las interrupciones deben ser servidas de manera que no imposibiliten de forma permanente el desarrollo de las tareas en proceso.
b) Debe haber algún método de permitir diversas interrupciones si existe más de un sistema que puede solicitarlas.
c) Si hay señales más importantes que otras o algunos periféricos requieren un servicio más rápido que otros, deberá existir un sistema jerárquico de interrupciones.

1.4.3.2 *Servicio de las interrupciones*

A partir del instante en que se solicita una interrupción, se desencadenan una serie de acciones, algunas de forma automática y otras que deben ser realizadas por el programa.

En general, siempre se termina de ejecutar la instrucción en curso, salvo en el caso de algunos microprocesadores que disponen de instrucciones que precisan un tiempo de ejecución largo, como las de multiplicación, división y transferencias de bloques de información, y permiten la interrupción en fase de ejecución.

Si la petición es aceptada, se inhibe total o parcialmente el sistema de interrupciones y se guarda el contador de programa en algún registro interno, en el stack o en una posición determinada de memoria. A continuación se pasa a la rutina de servicio en forma de salto vectorizado o bien se genera un ciclo especial de lectura a fin de obtener la próxima instrucción suministrada en este caso por el propio periférico.

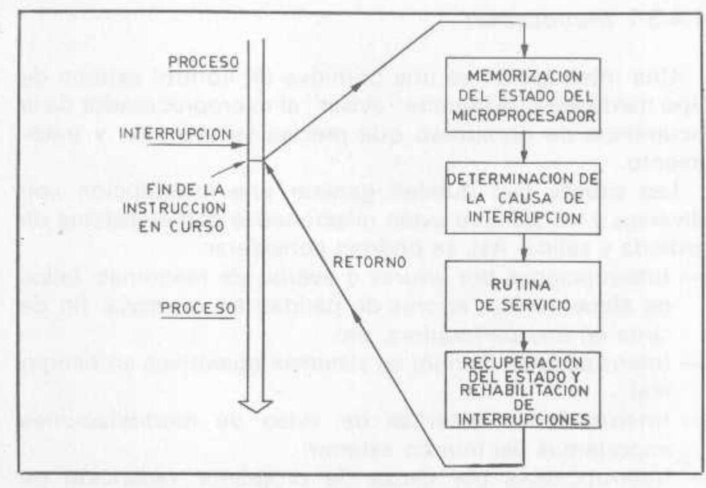


Figura 1.23 Diagrama de servicio de una interrupción.

A partir de este instante el programa toma el control, debiéndose realizar los siguientes pasos (fig. 1.23):

a) Guardar el estado de la tarea en curso: se deberán "salvar" todos aquellos registros que puedan ser modificados por la rutina de interrupción y que se precisen para poder reemprender la tarea interrumpida. En la mayoría de los microprocesadores se dispone de una estructura de stack para facilitar este tratamiento, que en algunos (6800) se realiza de una forma automática. En algunos casos se dispone de la posibilidad de efectuar un cambio (swap) de registros operativos.
b) Identificación de la fuente de interrupción si pueden existir más de un sistema o causas para solicitarla.
c) Rutina de servicio: ejecución de las tareas asociadas con el servicio de una interrupción concreta. Constituye el tiempo realmente útil de todo el tratamiento.
d) Restitución del estado de la tarea interrumpida.
e) Rehabilitación, en caso de que se precise hacerlo, del sistema de interrupciones.
f) Retorno al programa interrumpido.

1.4.3.3 *Control de transferencias*

El método de control de transferencias por interrupciones consiste, básicamente, en que sea el propio periférico el que avise a la CPU de su disposición para aceptar una información (salida) o de que dispone de datos para entregar al microprocesador (entrada).

En este caso, el microprocesador interrumpirá la ejecución de la tarea que estuviese realizando, dará servicio al periférico y continuará con la tarea que quedó suspendida. De esta forma se penaliza al mínimo el tiempo de proceso con trabajos de gestión de transferencias.

El control por interrupciones permite:

— Un mejor aprovechamiento del sistema en general.
— Una respuesta rápida a un suceso externo.
— Sincronizar el mundo exterior con el proceso.
— Un cierto nivel de descentralización, ya que son los usuarios (periféricos) los encargados de "mover" el sistema.

1.4.3.4 *Niveles y prioridades*

La posibilidad de disponer de varias líneas de interrupción presenta las ventajas siguientes:

— Facilidad de identificación de la fuente de interrupción.
— Asignación de prioridades de una forma implícita.
— Posibilidad de enmascaramiento parcial del sistema de interrupciones.

Muchos microprocesadores no disponen de un sistema jerárquico de interrupciones, por lo que la asignación de prioridades deberá realizarse a partir del software o de hardware externo:

1) *Software:* Se establece una cadena de consultas de estado a fin de conocer la fuente de la interrupción. De esta forma se establecen los niveles de prioridad que pueden ser fijos por el orden en que se consultan los diversos periféricos, o variables si se desarrolla un algoritmo de asignación dinámica de prioridades.

2) *Hardware:* Existe un sistema externo a la CPU que gestiona las prioridades e indica al microprocesador la fuente de interrupción. Por la forma de realizarlo se pueden distinguir algunas alternativas:

— Estructura encadenada (daisy-chain, fig. 1.24): cuando el microprocesador recibe una petición de interrupción, lanza una señal de reconocimiento que se propaga de un

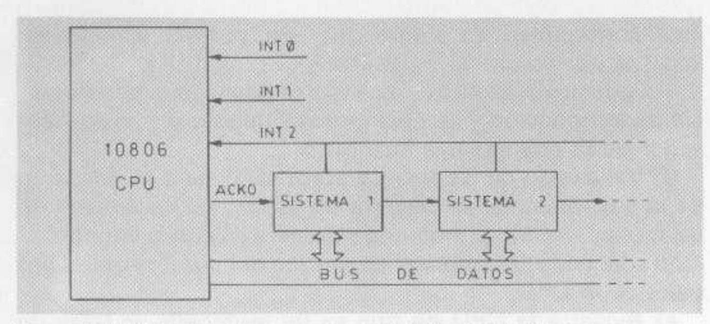


Figura 1.24 Estructura de interrupciones del Rockwell PPS-8.

periférico a otro. El primer sistema en la cadena que ha solicitado servicio bloquea la transmisión de esta señal y lanza, a la vez, código de identificación que indica al microprocesador la fuente de la petición más prioritaria.
— Controladores especializados. Son sistemas que reciben en paralelo todas las peticiones y se encargan de la gestión descentralizada de las mismas.

La estructura básica de un posible controlador (fig. 1.25) estaría formada por un sistema codificador, en este caso de 16 entradas, que suministrase el código correspondiente a la petición más prioritaria solicitada. Este código se compara con el nivel mínimo aceptable, programado por el microprocesador y, en caso de que la petición sea válida, se solicita la interrupción. Otra posible alternativa es que cada nivel fuese enmascarable de forma independiente. El microprocesador puede leer el estado a fin de conocer la causa de la interrupción.

Muchas familias de microprocesadores disponen de circuitos LSI programables especializados para las funciones de control de interrupciones (Intel 8259, Texas TMS 9901...). Estos controladores se tratarán en otros capítulos.

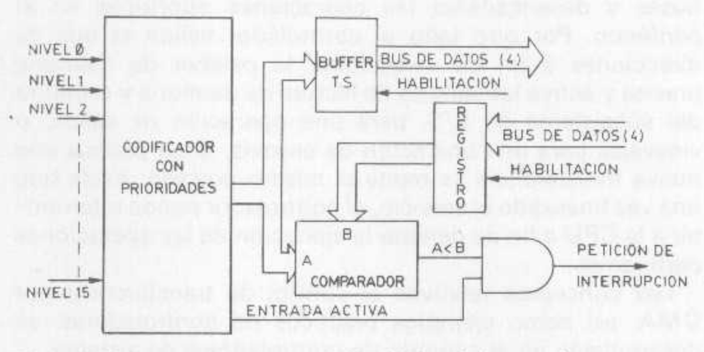


Figura 1.25 Esquema de un controlador básico de interrupciones.

1.4.3.5 *Estructuras de interrupción*

La potencia del sistema de interrupciones es muy variable al considerar distintos microprocesadores. Estas diferencias se presentan en:
— Número de entradas de interrupción de la CPU.
— Formas de reconocimiento de las interrupciones.
— Prioridad y modo de operación asociados a cada línea.
— Flexibilidad en la tarea de guardar y devolver el estado del programa interrumpido.
— Forma de obtención de la dirección de la rutina de servicio.
— Existencia de controladores programables compatibles que permitan aumentar la potencia y prestaciones del sistema de interrupciones.

A continuación se presentan algunos ejemplos de microprocesadores existentes en el mercado, en los que se pueden observar algunas alternativas distintas en cuanto a la concepción del sistema de interrupciones:

1) *Intel 8085* (fig. 1.7). El 8085 dispone de 5 líneas de petición de interrupción:

—Interrupciones vectorizadas: TRAP, RST7.5, RST6.5 y RST5.5 (por orden de prioridades decrecientes). Generan una llamada a subrutina vectorizada de forma directa a posiciones fijas de memoria. TRAP no es enmascarable y el resto lo son, de forma global o independientemente actuando sobre el registro de máscaras.

—INTR: Es la interrupción menos prioritaria y no dispone de una máscara particular sino que es habilitada o inhibida conjuntamente con todo el sistema de interrupciones. El proceso de servicio es del tipo del 8080 y consiste en la generación de un ciclo de búsqueda de instrucción especial, validándose la lectura por medio de la señal de reconocimiento $\overline{INTA}$, a fin de obtener el código de la próxima instrucción a ejecutar, suministrada por tanto por el propio periférico, y que en general consiste en una llamada a subrutina con direccionamiento implícito (Restart, 1 byte) o directo (Call, 3 bytes).

2) *Rockwell PPS-8* (fig. 1.24). El PPS-8 dispone de 3 líneas de petición de interrupción:

—INTO: Es la interrupción más prioritaria y no es enmascarable. Normalmente es utilizada para detectar fallos en la alimentación.

—INT1: Es enmascarable por software. Su utilización más característica es la de reloj de tiempo real.

—INT2: Es la interrupción menos prioritaria y es enmascarable. Normalmente se utiliza para comunicación con los periféricos, con los que se establece un sistema jerárquico de interrupciones por cadena hardware. Al ejecutar la instrucción de lectura del estado de interrupciones (RIS) se genera un pulso (ACKO) que se propaga de sistema en sistema, de forma que el primero de ellos que haya solicitado servicio bloquea la transmisión y coloca en el bus de datos un código determinado.

La activación de cualquiera de las interrupciones, no enmascaradas, ocasiona la inhabilitación del sistema y la llamada a subrutina de forma vectorizada indirecta, obteniéndose las direcciones de salto en unas posiciones predeterminadas de memoria.

3) *Texas 9900* (fig. 1.26). El 9900 dispone de una única línea de interrupción con 16 niveles asociados, de los que el nivel 0 (más prioritario) está reservado a la función de $\overline{Reset}$. El microprocesador dispone de un registro interno de estado que contiene la máscara de interrupción, gestionable por software, y con la que se compara el nivel de la interrupción que solicita servicio, sirviéndose ésta sólo si iguala o supera al nivel mínimo permitido.

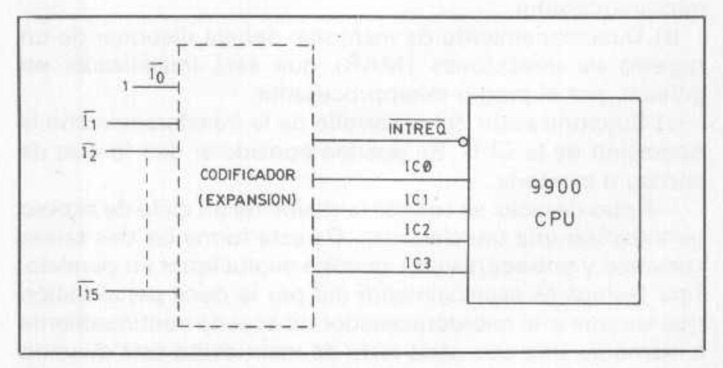


Figura 1.26 Bus de interrupciones del 9900.

La estructura de registros operativos en memoria de la que dispone el 9900 facilita el tratamiento subsiguiente a la aceptación de una interrupción: se establece un cambio de contexto, obteniéndose los nuevos valores del puntero de registros (Workspace Pointer, WP), del contador de programa (PC) de unas posiciones fijas de memoria asociadas a cada nivel de interrupción, y guardándose los antiguos WP, PC y ST (registro de estado) en los registros 13, 14 y 15 del nuevo banco de registros. La máscara de interrupciones se modifica de forma que sólo se acepten interrupciones de un nivel superior a la que es servida. Los registros son restituidos por medio de la instrucción de retorno de interrupción RTWP.

1.4.4 Sistema de acceso directo a memoria

Los métodos de transferencias programadas, aún disponiendo de un mecanismo de control rápido y eficiente como las interrupciones, presentan dos tipos de problemas:

1) Se puede penalizar de forma excesiva el tiempo dedicado a tareas de proceso, sobre todo si las necesidades de transferencia son muy frecuentes.

2) La máxima velocidad de transferencia obtenible puede no ser suficiente para las necesidades del periférico, especialmente en el caso de transferencias de información en bloques, como puede ser, por ejemplo, la lectura de un sector de un disco.

La técnica de acceso directo a memoria (DMA) consiste en enlazar directamente los periféricos con la memoria, de forma que la transferencia de información se hace de forma transparente a la CPU. Para este tipo de control será preciso que:

1) El microprocesador (CPU) se inhiba del gobierno de las líneas que serán utilizadas para el control de la transferencia. Para ello será necesario que se disponga de primitivas hardware de control para forzar al microprocesador a "congelar" su estado al término del ciclo de máquina en curso (no es preciso terminar la instrucción) y disponer las salidas de la CPU en estado de alta impedancia. Esta necesidad está motivada por la estructura de buses compartidos de que disponen los microcomputadores.

2) Exista un hardware externo de control para la gestión de la transferencia (controlador de DMA). Ha sido precisamente el desarrollo de circuitos integrados LSI especializados y compatibles con el resto del sistema lo que ha potenciado enormemente la utilización de esta técnica en el campo de los microcomputadores, anteriormente restringida a su aplicación en sistemas más complejos.

Algunos ejemplos son el Intel 8257, Rockwell 10817, RCA COSMAC (integrado en la CPU), Fairchild 3854...

Las funciones del controlador son:

a) Iniciar la transferencia por petición del periférico o del microprocesador.

b) Direccionamiento de memoria: deberá disponer de un registro de direcciones (MAR) que será inicializado, en general, por el propio microprocesador.

c) Sincronización del desarrollo de la transferencia con la operación de la CPU. Se pueden considerar dos formas de acceso a memoria:

—Robo de ciclo: se solicita la cesión de un ciclo de acceso para realizar una transferencia. De esta forma las dos tareas (proceso y entrada/salida) pueden evolucionar en paralelo. Esta técnica es especialmente útil por la poca penalización que supone si el microprocesador no accede continuamente a memoria, sino que cada ciclo de instrucción está dividido en dos subciclos (búsqueda y ejecución), o si la memoria está configurada de forma que sea posible simultanear accesos por programa y por DMA.

—Dedicación exclusiva: cuando se gestiona la transferencia de información y se para completamente el microprocesador hasta que se haya finalizado.

d) Validación de las señales apropiadas para la ejecución de la transferencia, es decir, la activación de las señales de escritura y lectura de memoria y periféricos con la temporización adecuada "simulando" la gestión del bus de control por parte de la CPU.

e) Avisar a la CPU de que se ha terminado la tarea de transferencia que se estaba realizando. En general, el controlador dispondrá de un contador, inicialmente programado por el microprocesador, y que se ve decrementado por cada transferencia elemental ejecutada, de forma que se puede indicar a la CPU, normalmente por interrupción, que o bien dispone de nuevos datos actualizados para su tratamiento (entrada por DMA) o bien puede reinicializar una nueva transferencia de salida por DMA.

f) Gestionar las prioridades oportunas si existe más de un canal con el que se puede establecer una comunicación por DMA e informar al microprocesador del canal al que se está sirviendo.

Un diagrama de bloques de la organización de un sistema para transferencia por DMA se presenta en la figura 1.27.

En el caso de que un periférico solicite un acceso directo a memoria (DMA REQ) y esta petición sea aceptada por el controlador, éste genera la señal de petición de DMA al microprocesador (HOLD). Una vez finalizado el ciclo de instrucción o el ciclo de máquina en curso, según el microprocesador considerado, éste activa la señal de reconocimiento (HOLD ACK) que indica al controlador que puede tomar el gobierno de los buses. El controlador activa la señal de cesión de servicio (DMA ACK), a fin de seleccionar el periférico estableciendo su conexión a los buses y desencadenar las operaciones oportunas en el periférico. Por otro lado el controlador valida el bus de direcciones a fin de seleccionar la palabra de memoria precisa y activa las señales de lectura de memoria y escritura del subsistema de E/S, para una operación de salida, o viceversa para una operación de entrada. Si se precisa una nueva transferencia se repite el mismo proceso, hasta que una vez finalizado el servicio, el controlador puede interrumpir a la CPU a fin de generar la ejecución de las operaciones pertinentes.

Los conceptos relativos al control de transferencia por DMA, así como ejemplos prácticos de controladores, se desarrollarán en el capítulo de controladores de sistema.

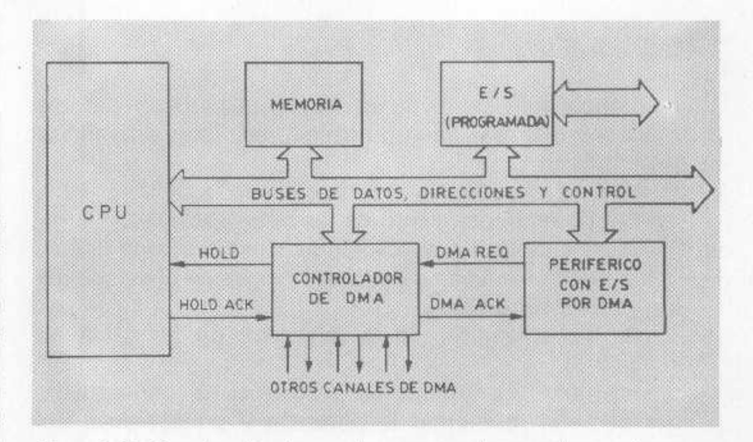


Figura 1.27 Organización de un microcomputador con sistema de acceso directo a memoria.

1.5 TIPOS DE PERIFÉRICOS

Los periféricos son utilizados en un sistema microcomputador para establecer la comunicación con el mundo exterior. Debido al incremento en los tipos de aplicaciones en los que se utilizan y a la evolución tecnológica que lo ha permitido, el universo de los periféricos se ha ido desarrollando con gran rapidez, en parte motivado precisamente por las necesidades crecientes de disponer de periféricos aptos para su utilización en sistemas basados en microprocesadores.

Al considerar el tema de los periféricos adecuados para microcomputadores, cobran una mayor importancia algunas de sus características, como:

a) *Coste:* En muchos casos el coste de un periférico y de la circuitería asociada para su control puede representar una parte muy significativa del coste total del sistema, e incluso ser superior a todo el resto, por lo que se hace más importante todavía el poder disponer de periféricos baratos y sistemas de control sencillos, o la disponibilidad de circuitos LSI especializados para el control del periférico. De otra forma es muy posible que no quede justificada la utilización de un microcomputador, ya que se conseguirían rendimientos mayores con un procesador más potente con un incremento de coste marginal poco significativo.

b) *Tamaño y peso:* Muchos sistemas basados en microcomputadores (terminales, equipos de instrumentación) son portátiles, por lo que el volumen y peso del equipo es un factor determinante.

c) *Consumo:* Característica bastante ligada al tamaño y peso del equipo por el consumo asociado a las partes electromecánicas del mismo. Puede tener mucha importancia a la hora de diseñar las fuentes de alimentación, compactas y baratas.

d) *Fiabilidad:* Por la forma de operación de muchos sistemas microcomputadores, en comunicación continua con el mundo real, es importante disponer de equipos periféricos con índices de fiabilidad lo más próximos posibles al del propio microcomputador.

Los tipos de periféricos utilizados en sistemas microcomputadores son muy variados en función de la aplicación a la que son destinados, haciéndose difícil una enumeración exhaustiva de todos ellos. En la figura 1.28 se pretende hacer una clasificación bajo el criterio del elemento dialogante con el microcomputador.

En capítulos posteriores se hará un tratamiento detallado de la mayoría de ellos y de las formas de interconexión con el microcomputador, por lo que en este apartado se considerarán sólo a título de clasificación y presentación:

1) *Periféricos internos:* Son subsistemas del microcomputador que no son en realidad periféricos, aunque pueden ser considerados como tales. Entre éstos se podrían destacar los circuitos utilizados para aumentar la potencia aritmética del sistema, como multiplicadores monolíticos y generadores de funciones, y los temporizadores programables que permiten la realización por hardware tales como el contaje de sucesos externos, gestión de un reloj diario, reloj de tiempo real, etc.

2) *Interruptores, pulsadores y conmutadores:* Permiten entrar datos al sistema, seleccionar diversas opciones o modos de funcionamiento, lanzar órdenes de actuación, de una forma manual. La forma de establecer el contacto depende del tipo considerado: contactos metálicos, de lámina, acoplamiento óptico, efecto capacitivo, magnéticos, por efecto Hall, etc.

3) *Teclados:* Es una forma de disponer de un conjunto de pulsadores, de forma compacta, para entrar información alfanumérica y de control al sistema. Presenta ventajas en cuanto a flexibilidad de operación, sencillez de manejo, coste y estética se refiere.

4) *Visualizadores:* Dan al operador de una forma visible información sobre el sistema, pudiéndose realizar por diversos métodos ópticos (diodos emisores de luz, cristal líquido, incandescencia, fluorescencia) y electromecánicos. La información puede ser de tres tipos, según los visualizadores sean:

— binarios: indicadores individuales;
— numéricos: permiten visualizar los dígitos de 0 al 9 y algunas letras;

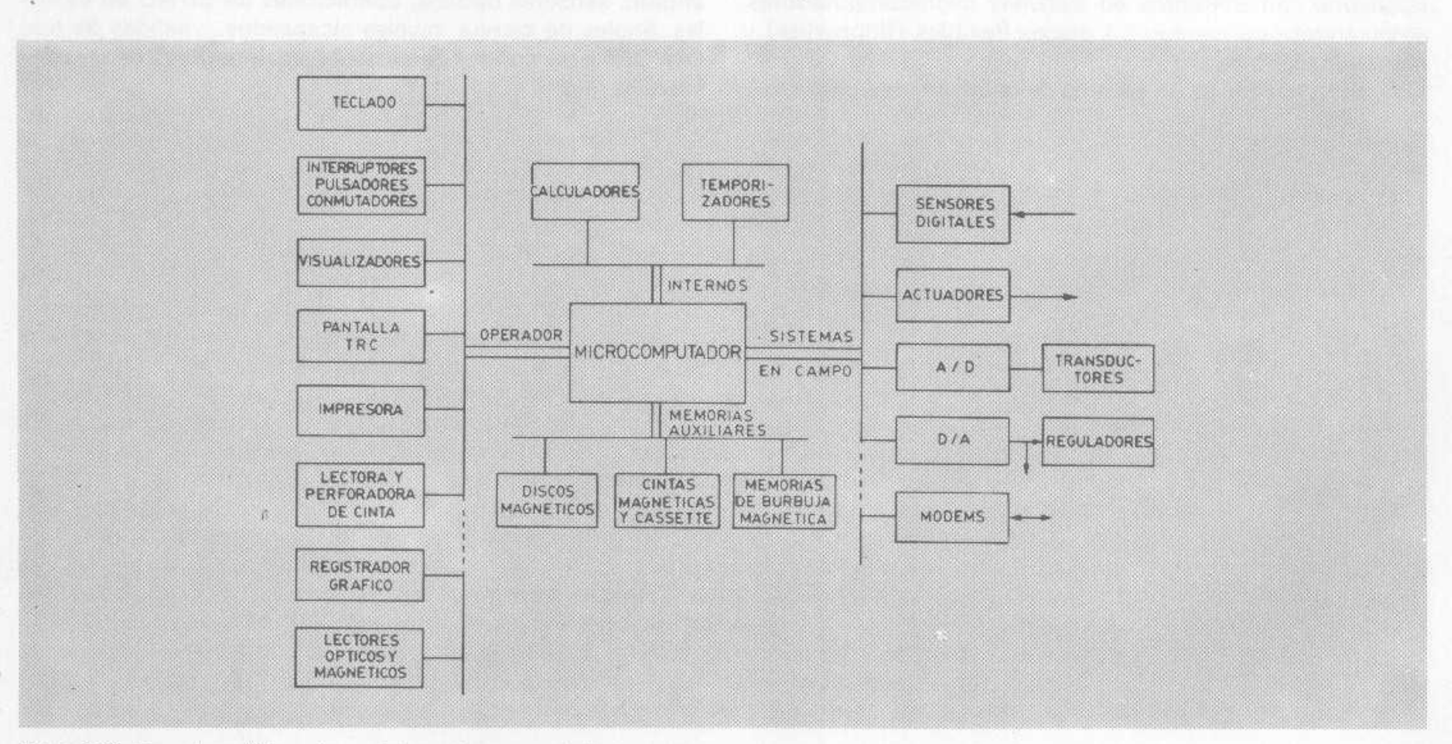


Figura 1.28 Tipos de periféricos de un sistema microcomputador.

— alfanuméricos: permiten visualizar todo el alfabeto y caracteres especiales.

Por la forma de control pueden ser estáticos (información fija) y multiplexados (refresco periódico).

5) *Pantalla TRC:* Consiste en la incorporación de un sistema de control a un tubo de rayos catódicos para disponer de un elemento de visualización alfanumérica. Un terminal TRC está constituido por una pantalla y teclado y el hardware/software preciso para que su utilización sea flexible y sencilla. Existen sistemas más elaborados que permiten la visualización de gráficos, pantalla en color o entrada de datos por pantalla por medio de un puntero luminoso (light-pen).

6) *Impresoras:* Las impresoras más utilizadas en sistemas microcomputadores son las de media y baja velocidad, numéricas o alfanuméricas y, en general, de escritura carácter a carácter. Para aplicaciones que precisen la escritura rápida de largos listados se puede disponer de impresoras de línea. Por el modo de escritura pueden ser electromecánicas (de tambor, de cadena, de bola, de margarita, de matriz de agujas), térmicas, etc. Algunas impresoras permiten también la generación de gráficas por puntos o continuas (registradores y plotters).

7) *Lectoras y perforadoras de cinta:* Son periféricos que permiten la lectura y escritura de información, tomando como soporte físico una cinta perforada de papel o plástico. Son periféricos relativamente lentos, con velocidades de perforación de hasta 300 caracteres/s y lecturas del orden de 1000 c/s para lectoras ópticas y de efecto capacitivo.

Un tipo de periférico muy utilizado es el teleimpresor, siendo uno de los más conocidos el Teletipo ASR-33 (de Teletype Corporation), el cual dispone en un único equipo de una impresora de caracteres, cónsola de teclado, lectora y perforadora de cinta. Es un periférico de baja velocidad (10 c/s) y la comunicación con el procesador se realiza en serie con un canal asíncrono a 110 bauds.

8) *Memorias auxiliares (de masa):* Permiten disponer de un medio de memoria de gran capacidad a bajo coste. De los sistemas habituales, los discos rígidos y cintas magnéticas raramente son utilizados en sistemas microcomputadores, empleándose en cambio los discos flexibles (floppydisk) y cassettes digitales.

El disco flexible es un sistema de relativamente bajo coste y bastante fiables. Los dos formatos utilizados son los de 5" (mini-floppy) y 8" (estándar). Existen transportes de disco que utilizan una o dos caras del disco, densidad de grabación doble o sencilla y densidades de pista doble y simple, dando como resultado sistemas de capacidades muy diversas. La capacidad de un disco flexible estándar de simple cara y simple densidad oscila entre 2 y 3 Mbits y la velocidad de transferencia, una vez posicionado, es del orden de 250 Kbits/s.

Los cassettes digitales son más baratos que el disco flexible, en coste/bit, pero tiene desventajas en cuanto a tiempo de acceso, velocidad de transferencia e índices de fiabilidad. Se han desarrollado algunos sistemas de cartuchos magnéticos que permiten una mayor capacidad y velocidad que el cassette.

9) *Modems:* Son sistemas para la conversión y reconversión de señales digitales en analógicas, de forma que se pueda enviar la información, en serie, a largas distancias por medio de líneas de transmisión. Será por tanto preciso el disponer de un sistema modulador/demodulador (modem) en cada terminal de línea. Las técnicas más utilizadas son las de modulación de amplitud, frecuencia (FSK) y fase con referencia fija o diferencial (PSK). La transmisión puede efectuarse en modo full-duplex (transmisión y recepción simultáneas) o half-duplex.

10) *Convertidores A/D y D/A:* Muchas de las variables precisas en un sistema de control (temperatura, presión, posición, velocidad, caudal, insolación) son captadas por transductores que trasladan la magnitud física considerada en variables eléctricas continuas (tensión, corriente), por lo que es preciso un medio de conversión A/D de la información de forma que pueda ser procesada por el ordenador. Igualmente, la información de salida puede ser necesaria en forma analógica, precisándose en este caso de un convertidor D/A.

11) *Sensores y actuadores digitales:* Son sistemas que, o bien por la forma de captación (entrada) o de actuación (salida), interaccionan con el microcomputador en forma digital. Algunos ejemplos podrían ser: codificadores de ángulo, sensores ópticos, condiciones de SI/NO en válvulas, finales de carrera, niveles alcanzados..., salidas de tipo ON/OFF para control de motores, accionadores de electroválvulas, etc.

protocolos de comunicación y sincronización

2

Miguel García Hoffmann

2.1 INTRODUCCIÓN

2.1.1 Distribución de funciones en los sistemas informáticos

Una de las características claramente observable en la evolución de la tecnología de los computadores es la tendencia a la modularidad. Los elementos estructurales de los computadores se conciben cada vez más como unidades dotadas de cierta autonomía que cooperaran entre sí. Esta tendencia se fundamenta no sólo en la búsqueda de diseños más rápidos y eficientes sino también en el principio de la división de funciones que facilita la concepción, diseño y mantenimiento de los diversos elementos que forman el sistema.

También es fácil constatar la atención que se ha prestado en la última década al estudio de los sistemas informáticos distribuidos que brindan la posibilidad de compartir recursos informáticos y aumentan la fiabilidad y disponibilidad de los sistemas a un precio justificable. Los multiprocesadores y las redes de computadores son ejemplos de este tipo de sistemas. La aparición y extensión de los microprocesadores incrementan el interés en los sistemas distribuidos al posibilitar la concepción de redes de microprocesadores y de nuevas arquitecturas multimicroprocesador formadas por un número elevado de microprocesadores que se comuniquen entre sí de forma bien definida, sin necesidad de un control centralizado.

2.1.2 Comunicación y protocolos

Estas consideraciones sugieren la posibilidad de contemplar todo sistema informático como un conjunto de unidades más o menos autónomas con una función bien definida que colaboran entre sí para la consecución de tareas determinadas. La naturaleza y los objetivos perseguidos con esta cooperación difieren radicalmente de un sistema a otro, existiendo, sin embargo, ciertos aspectos comunes a todos ellos. Desde la óptica en que nos hemos situado puede señalarse como aspecto común de todos los sistemas la necesidad de intercambiar información entre los elementos que lo integran. Estos pueden ser circuitos, modems, concentradores, terminales, computadores, procesos, personas, etc.

El intercambio de información entre los componentes de un sistema se denomina comunicación. La forma en que se realiza la comunicación depende de múltiples factores pero, en cualquier caso, es indispensable establecer claramente las reglas que han de seguirse en el intercambio de información. Se denomina protocolo de comunicación al conjunto de reglas que siguen la comunicación entre los elementos de un sistema. La materialización hardware o software de estas reglas recibe asimismo la denominación de protocolo.

Obsérvese que en estas definiciones no se ha impuesto ninguna restricción al tipo de información que intercambian los elementos del sistema. El concepto de comunicación se extiende desde el intercambio de datos, que presumiblemente involucrará procesos complejos de software, hasta la simple señal que notifica a un elemento el estado de otro.

2.1.3 Concurrencia

En general, la acción de los diversos elementos discurre simultáneamente produciéndose la comunicación en momentos determinados. La concurrencia es la piedra angular del incremento de velocidad de los sistemas informáticos. Considérese, por ejemplo, la evolución de la forma de cooperar la unidad central de procesos, la memoria y la unidad de entrada/salida de los computadores en el caso de un periférico de entrada que suministra información para almacenar en memoria.

Originalmente la unidad central de procesos detenía su acción al intentar ejecutar una sentencia de entrada hasta que el periférico estuviese en condiciones de atenderla. Dada la extraordinaria lentitud de los periféricos respecto a la unidad central, esta forma de proceder supone una pérdida notable de tiempo. Para resolver este problema se concibió un sistema de transferencia, consistente en no ejecutar una instrucción de entrada hasta que el periférico esté dispuesto a realizar la transferencia. Esta forma de comunicación exige que la unidad central de procesos conozca el estado del periférico, para lo cual se incluyen en el programa instrucciones de prueba de estado del periférico. Si éste se encuentra realizando alguna operación, la unidad central continúa la ejecución del programa, simultaneando su acción con la del periférico. Una tercera forma de plantear la transferencia, mucho más práctica que las dos anteriores, consiste en hacer que el periférico interrumpa a la unidad central cuando esté disponible, de modo que ésta ejecute un programa específico destinado a realizar la transferencia. Para evitar la intervención de la unidad central en cada transferencia se concibieron y crearon los canales, unidades capaces de transferir bloques de información entre memoria y el sistema de E/S sin más intervención de la unidad central que la necesaria inicialización del canal. Un paso más en la liberación de la unidad central en las tareas de gestión de las transferencias lo constituyen los canales capaces de almacenar y ejecutar un programa de control de las transferencias que propiamente ha sido generado por la unidad central. En este caso el canal puede considerarse propiamente como un procesador de E/S cuya actividad discurre paralelamente a la de la unidad central.

Este breve ejemplo sirve para ilustrar la evolución en un caso particular de la arquitectura de los computadores, tendente a la separación de funciones que conduce finalmente a una estructura modular en la que los diferentes elementos realizan sus tareas concurrentemente, comunicándose tan sólo cuando es preciso. Obsérvese que en cualquiera de las formas de transferencia citadas existe la comunicación entre la unidad central y la de E/S, si bien las reglas que la rigen son cada vez más complejas. En cualquier caso, el protocolo de comunicación juega un papel decisivo en el correcto funcionamiento del sistema en su conjunto.

2.1.4 Sincronización

El intercambio de información, la comunicación, debe realizarse en momentos determinados y con la aquiescencia de los elementos que en ella participan. Este consentimiento puede estar implícito en la naturaleza del proceso o no, siendo necesario en este caso disponer de mecanismos que sincronicen las acciones de los procesos que conducen a la comunicación.

Considérese el caso de un computador que controla un panel de indicadores luminosos a través de una matriz de biestables externos. La actualización de la información de los biestables se realiza en el momento en que el computador lo decide unilateralmente: la matriz de biestables es un elemento pasivo y la comunicación no requiere sincronización previa. Por el contrario, en cualquiera de los modos de transmisión considerados en el ejemplo anterior del periférico de entrada, se ha supuesto la necesidad de sincronizar la operación de la unidad central de procesos y la de E/S.

La sincronización va implícita en el concepto de protocolo puesto que es parte de las reglas de intercambio. Es interesante observar que en muchos casos el objeto de la comunicación es exclusivamente la sincronización.

2.1.5 Interfaces microprocesador-periférico

La conexión de los microprocesadores a los periféricos se realiza usualmente a través de elementos hardware especializados, capaces de controlar el intercambio de información entre el microprocesador y el exterior. En general estos elementos realizan dos protocolos; uno que rige el intercambio de información entre el microprocesador y el propio elemento y otro encargado de la comunicación entre éste y el periférico. Los principios de funcionamiento de estos circuitos y de los protocolos que realizan se tratan ampliamente en la segunda parte de este libro, por lo que en este capítulo centramos la atención en la problemática de los protocolos de sistemas distribuidos que tanta importancia revisten para el desarrollo de la microinformática en particular y de la informática en general.

2.2 SISTEMAS DISTRIBUIDOS Y COMUNICACION

2.2.1 Introducción

Los sistemas distribuidos son aquéllos que tienen cada uno de sus subsistemas básicos (proceso, control y memoria) físicamente distribuidos en mayor o menor grado. Los multiprocesadores son un caso particular de sistemas distribuidos en los que solamente se exige la distribución de la unidad de proceso, pudiendo encontrarse la memoria y el control distribuidos o no. Los problemas de la comunicación entre los distintos módulos de los multiprocesadores son muy peculiares y su estudio detallado escapa del contexto de este libro. El lector interesado en este tema puede consultar las referencias [17, 25, 26, 27, 36, 37, 39, 42].

Aún cuando en el apartado anterior se ha empleado el término protocolo en un sentido generalizador que engloba la comunicación entre elementos hardware de un mismo sistema, es necesario indicar que la literatura especializada suele reservar la denominación de protocolo al campo de la informática distribuida y en particular al de las redes de computadores.

Desde el inicio de la construcción de la red Arpanet a cargo del Departamento de Defensa de los EE.UU., hacia 1967, hasta nuestros días, se ha hecho un esfuerzo considerable en el estudio de cómo deben concebirse y realizarse los protocolos necesarios para el funcionamiento de las redes de computadores. El aprendizaje en la construcción de las primeras redes experimentales (Arpanet-1967 [30], EIN-1971 [14], Cyclades-1973 [29], NPL-1970 [31]) ha revertido en un crecimiento impresionante del número de redes públicas y privadas operativas, en construcción o planeadas [8, 12, 18, 19, 38].

Un criterio ampliamente extendido de clasificación de las redes se acomoda a las características y componentes de la comunicación [41]. Según este criterio las redes pueden dividirse en redes para compartir recursos, redes para cálculo distribuido y redes para comunicaciones remotas.

En el primer tipo de redes el objetivo perseguido es permitir el acceso y utilización remota de recursos informáticos tales como ficheros en disco, impresoras, etc. Para conseguir esto es necesario poder establecer la comunicación entre un programa que se ejecuta en un computador y el programa que controla en otro la utilización del recurso. En las redes orientadas al cálculo distribuido como son los procesadores paralelos, algunos sistemas multiprocesador especializados y los sistemas distribuidos de control en tiempo real, la comunicación se establece entre programas localizados en diferentes computadores que cooperan en una tarea común. Finalmente, las redes para comunicaciones remotas ofrecen conexiones entre usarios y sistemas de cálculo remotos.

En los tres tipos de redes los distintos elementos del sistema distribuido (programas, procesos, periféricos, terminales interactivos, etc.) intercambian información, unas veces a través de mensajes cortos y relativamente independientes los unos de los otros (p. ej., transacciones de terminales) y, otras, mediante secuencias de mensajes de mayor longitud conteniendo información relacionada (p. ej., transferencias de ficheros). En cualquier caso es necesario disponer de protocolos capaces de controlar la transferencia de información entre los elementos del sistema.

2.2.2 Jerarquía de la comunicación

Originalmente los protocolos se diseñaban como un proceso único que controlaba todas las operaciones de la red destinadas a la comunicación. El incremento en el número de redes y la diversificación de servicios que deben suministrar exige la creación de normas para la confección de protocolos. De este aspecto se han encargado fundamentalmente el «International Telegraph and Telephone Consultative Commitee» (CCITT), la «International Organization for Standarization» (ISO), el «American National Standards Institute» (ANSI) y la «International Federation for Information Processing» (IFIP) en un intento de crear normas de aceptación internacional.

El criterio adoptado por estos organismos —y por el trabajo actual en general— concuerda con una concepción

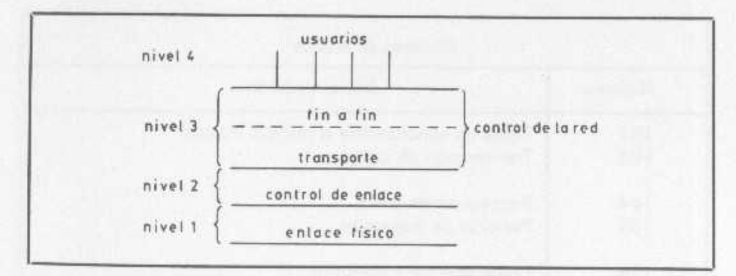


Figura 2.1 Niveles de la jerarquía

de los protocolos como estructuras multinivel jerarquizadas. En ella, cada nivel crea un canal virtual que ofrece posibilidades de comunicación de características determinadas entre entidades concretas. Para un nivel determinado los niveles inferiores son transparentes, es decir, éstos le ofrecen un conjunto de funciones de comunicación que él utiliza sin tener que tomar en cuenta la forma en que aquélla se realiza.

La delimitación exacta de los niveles y la determinación de qué entidad debe realizarlos depende del tipo de sistema distribuido y es motivo de polémica. La figura 2.1 muestra la división en niveles más extendida. Entre las ventajas que comporta esta visión jerarquizada se cuentan la mayor sencillez en la concepción de los mecanismos de comunicación al separar las distintas funciones, la distribución entre los distintos elementos de las tareas de la comunicación y la mayor facilidad para realizar modificaciones puesto que, teóricamente al menos, la modificación de un nivel no debe influir en los otros.

Los protocolos del nivel 1 especifican las características físicas y eléctricas, así como los procedimientos empleados en el control del canal físico. Sobre este nivel se construyen los protocolos de enlace que controlan la comunicación sobre un enlace físico entre entidades, creando un canal virtual prácticamente libre de errores de comunicación. El nivel 3 lo constituyen los protocolos destinados a gobernar el flujo de información entre las entidades origen y destino de la comunicación. Se distinguen en ocasiones dos subniveles del nivel 3, el de transporte y el subnivel fin a fin, también denominado en algunas estructuras "host a host". El subnivel de transporte es el encargado de la transferencia de mensajes entre dos nudos origen y destino de la red de comunicaciones. En algunos sistemas este subnivel provee funciones de corrección de errores y control de flujo mientras que en otros es simplemente un mecanismo que transfiere información dejando este tipo de funciones al subnivel superior. El subnivel fin a fin es el responsable de controlar la transferencia de información entre los usuarios conectados a los nudos origen y destino de la red de comunicaciones. Para ello utiliza el canal virtual que ofrece el subnivel de transporte. Por último, los protocolos empleados por programas, terminales, usuarios, etc., con el fin de beneficiarse de los servicios de la red de comunicaciones, se denominan protocolos para compartir recursos y constituyen el cuarto nivel y superiores.

La delimitación exacta de estos niveles depende en parte del sistema de que se trate. La figura 2.2 muestra los diferentes niveles de una arquitectura típica. En apartados sucesivos se discuten con cierto detalle las funciones específicas de cada nivel.

2.2.3 Protocolos e interfaces de comunicación

Obsérvese que un protocolo determinado controla el flujo de información entre las dos partes físicamente separadas de

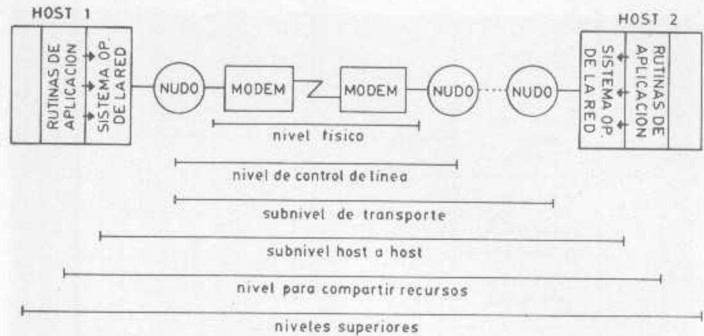


Figura 2.2 Diferentes niveles en una arquitectura típica

un mismo nivel. Ahora bien, igualmente necesario es establecer las reglas de comunicación entre los distintos niveles. Para distinguir los protocolos "horizontales" (entre partes de un mismo nivel) de los "verticales" (entre niveles adyacentes) se denomina a estos últimos interfaces de comunicación. La utilización del término interfaz es coherente con el hecho de que generalmente los equipos que realizan los protocolos de distinto nivel son diferentes, constituyendo las reglas de comunicación una verdadera interfaz. Contrariamente, la comunicación dentro de un nivel suele realizarse entre entidades idénticas o cuando menos similiares.

Antes de pasar a estudiar cada uno de los niveles es conveniente comprender las funciones básicas de los protocolos e interfaces:

Direccionamiento: la especificación del origen y destino de la información.

Control de error: la detección y recuperación de errores en la transmisión sobre el canal virtual de comunicaciones que se utiliza. En general todo protocolo ha de garantizar la transmisión correcta de una única copia de cada mensaje.

Control de flujo: se engloban en esta función las operaciones destinadas al mantenimiento del flujo de la información como son la selección de la ruta que han de seguir los mensajes, la reserva de espacio (de memoria) en la estación destino para información que ha de enviarse, etc.

Sincronización: todo protocolo ha de ser capaz de mantener en sincronismo las partes de un mismo nivel en el curso de la comunicación.

Otras funciones características de los protocolos son la conexión y desconexión de circuitos físicos en ciertas aplicaciones, el mantenimiento de una secuencia ordenada de los mensajes, la segmentación de mensajes en unidades de transmisión más pequeñas (paquetes) y su ulterior reconstrucción, el tratamiento de mensajes con prioridades diversas, etc.

Los protocolos de nivel 4 y superiores se realizan generalmente sobre máquinas de potencia media o grande [9, 22, 28, 33] de forma que los niveles que más interesa considerar aquí —los susceptibles de realizaciones con microprocesadores— son los niveles 2 y 3, correspondientes a los protocolos de control de línea y control de red, así como las interfaces de comunicación entre los diversos niveles.

2.3 EL ENLACE FÍSICO

2.3.1 Introducción

Siempre que dos entidades físicamente separadas han de intercambiar información es necesario disponer de un enlace

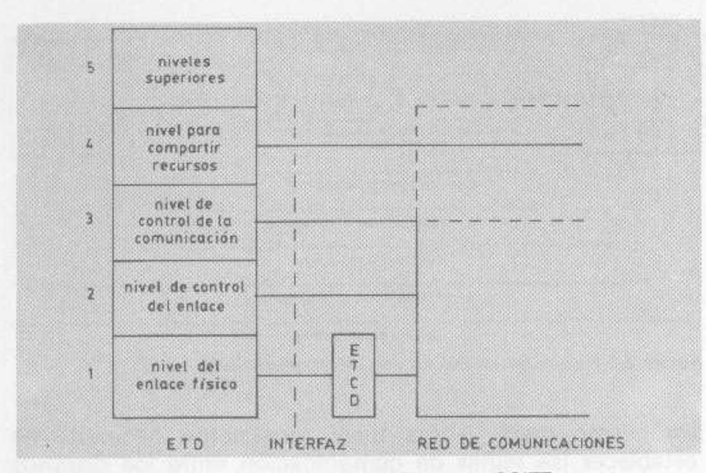


Figura 2.3 Niveles en la conexión ETD/ETCD según CCITT

físico y de algunas reglas que controlen su utilización. Es, por ejemplo, el caso de la comunicación a través de modems de los nudos de una red de comunicaciones o de las interfaces entre la red y sus usuarios. Este segundo caso se referencia en muchas ocasiones como la conexión entre equipos terminales del circuito de datos (ETCD) y equipos terminales de datos (ETD) (fig. 2.3.)

El enlace físico de comunicaciones lo constituyen los hilos eléctricos, canales ópticos, ondas electromagnéticas, etc. Las especificacions del nivel 1 crean una interfaz entre el enlace físico y el hardware de los elementos que han de utilizarlos [16]. A continuación se hace referencia a algunas normas CCITT, ISO y EIA que especifican las características funcionales, eléctricas y mecánicas de este nivel, remitiendo al lector para más detalles a las recomendaciones CCITT para la transmisión de información sobre la red telefónica (serie V) [5] y a las referentes a las redes públicas (serie X) [6].

2.3.2 Características funcionales

La recomendación X.24 sitúa la interconexión entre ETD y ETCD en un conector y define un conjunto de funciones para el intercambio de información: dos circuitos de datos (T y R), dos de control (C e I), dos de sincronismo (S y B) y tres

Denominac. del circuito de enlace	Nombre del circuito de enlace	Datos: Del ETCD	Datos: Hacia el ETCD	Control: Del ETCD	Control: Hacia el ETCD	Temporización: Del ETCD	Temporización: Hacia el ETCD
G	Tierra de señalización o retorno común						
Ga	Retorno común del ETD				X		
Gb	Retorno común del ETCD			X			
T	Transmisión		X		X		
R	Recepción	X		X			
C	Control				X		
I	Indicación			X			
S	Temporización para los elementos de señal					X	
B	Temporización para los multibitios					X	

Figura 2.4 Circuitos de enlace de las redes de datos

Circuito de enlace	
Número	Denominación
102	Tierra de señalización o retorno común
103	Transmisión de datos
104	Recepción de datos
105	Petición de transmitir
106	Preparado para transmitir
107	Aparato de datos preparado
108/1	Conecte el aparato de datos a la línea
108/2	Terminal de datos preparado
109	Detector de señales de línea recibidas por el canal de datos
125	Indicador de llamada
126	Selección de la frecuencia de transmisión

Figura 2.5 Denominación de los circuitos de enlace para modems de 200 baudios

de retorno común (G, Ga, Gb), tal como se indica en la figura 2.4.

La serie V de recomendaciones especifica las características funcionales que han de cumplir los modems para la transmisión de datos sobre la red telefónica. Así, las recomendaciones V.10 a V.31 tratan de las interfaces y modems para la banda de frecuencias vocales, mientras que la V.35 y V.36 se refieren a los modems de la banda ancha (60 a 108 kHz). A título de ejemplo la figura 2.5 lista los circuitos de enlace para modems de 200 b utilizados en la red telefónica (V.21).

2.3.3 Características eléctricas

Dos son las recomendaciones de la CCITT sobre las características eléctricas entre ETD y ETCD, la X.26 y la X.27.

La X.26, también desginada V.10, es similar a la norma EIA RS-423 y, estando orientada a equipos que emplean tecnología de circuitos integrados, ha demostrado ofrecer un rendimiento superior que la tradicional V.28 (EIA RS-232C) para tecnología de componentes discretos. El circuito de

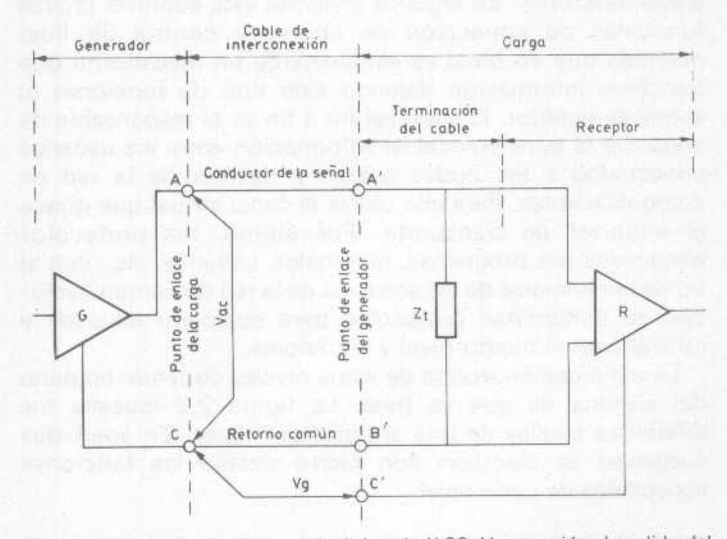


Figura 2.6 Circuito de enlace asimétrico de X.26. V_{ac}= tensión de salida del generador; V_g= diferencia de potencial entre tierras; A= punto activo de enlace del generador; C= punto de retorno común del generador; A'= punto activo de enlace de la carga; B'= punto de retorno común de la carga; C'= punto de referencia cero voltios del receptor

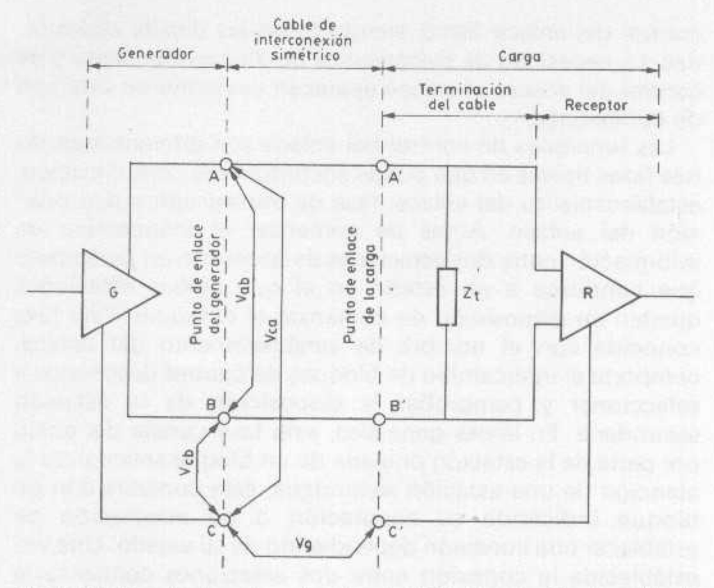


Figura 2.7 Circuito de enlace simétrico de X.27. V_{ab} = tensión de salida del generador entre los puntos A y B; V_{ca} = tensión del generador entre los puntos C y A; V_{cb} = tensión del generador entre los puntos C y B; Z_t = impedancia de terminación del cable; V_g = diferencia de potencial entre tierras; A, B, y A', B' = puntos de enlace; C, C' = puntos de enlace "referencia cero voltios"

transmisión establecido en X.26 consiste en un generador de una salida no compensado y un receptor diferencial compensado conectados por una señal única y un circuito común de retorno (fig. 2.6.)

La recomendación X.27 o V.11 define las características eléctricas de los circuitos de enlace simétricos de doble corriente para uso general con equipos de circuitos integrados. Esta recomendación es similar a la EIA RS-422. El transmisor es un generador diferencial y las características del receptor son idénticas a las de X.26. Esta conexión permite la transmisión de hasta 10 Mb/s (fig. 2.7).

2.3.4 Características mecánicas

La conexión de los equipos se realiza a través de un conector de 15 patas según establece la norma ISO DIS4903, que corresponde al conocido conector EIA RS-232C.

2.3.5 Algunas interfaces específicas

La recomendación X.20 especifica la interfaz entre el equipo terminal de datos y el equipo terminal del circuito de datos para servicios arrítmicos (start/stop) de transmisión en las redes públicas de datos. La figura 2.8 muestra la conexión empleada por X.20. Como puede apreciarse, utiliza

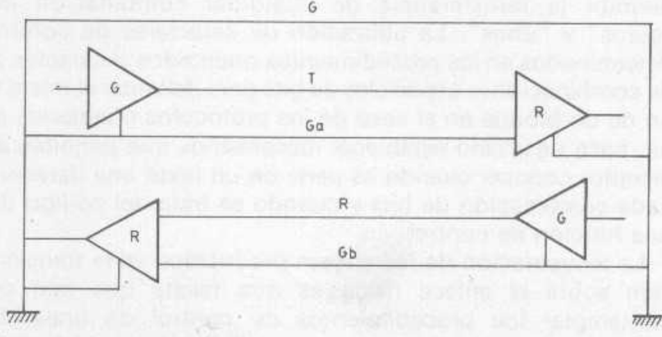


Figura 2.8 Conexión X.20 con ETD y ETCD X.26

parte de las funciones definidas por X.24 y las características eléctricas de X.26 tanto para el ETD como para el ETCD (el ETD puede también utilizar la X.27). Esta recomendación señala la forma de establecer un circuito de comunicación orientado al carácter que emplea el alfabeto V.3, esencialmente el mismo que el ASCII. Cada una de las estaciones que realiza la interfaz puede encontrarse en una de cuatro fases: dormida, establecimiento de la conexión, fase de datos y desconexión. La recomendación, que está pensada para la transmisión arrítmica de caracteres tanto en conexión punto a punto como en conexión multipunto, suministra diagramas de estados para las diferentes fases de la conexión. Estos aspectos corresponden, sin embargo, a los niveles 2 y 3 de la jerarquía de la comunicación. La recomendación X.20 bis es similar a la X.20, siendo compatible con los modems definidos por V.21.

Para crear interfaces entre ETD Y ETCD en funcionamiento síncrono sobre redes públicas de datos se ha creado la recomendación X.21 (fig. 2.9). Es ésta la recomendación más generalizada para la transmisión síncrona. Las funciones definidas en X.24 y las características eléctricas de X.26, y opcionalmente X.27 para ETD, son la base de esta interfaz. Opera en las mismas cuatro fases que el X.20 y está orientado al carácter empleando asimismo el alfabeto V.3. La recomendación X.21 bis hace referencia a la utilización en las redes públicas de datos de ETD y ETCD concebidos para su conexión con modems síncronos de la serie V.



Figura 2.9 Conexión X.21 entre ETD/ETCD

2.4 PROTOCOLOS E INTERFACES DE CONTROL DE LÍNEA

2.4.1 Generalidades

La finalidad de los protocolos e interfaces de control de línea (nivel 2) es el establecimiento sobre un enlace físico existente (nivel 1) de un canal secuencial que permite la comunicación entre las estaciones conectadas proveyendo funciones de correción de errores y de control del flujo de la información. Su misión es, por tanto, ofrecer a los programas, terminales, protocolos de nivel superior, etc., un canal virtual capaz de transmitir secuencialmente una única copia de cada mensaje a través de un medio (el físico) que por su naturaleza puede producir errores en la transmisión. Recordamos que la diferencia entre protocolo e interfaz de comunicación radica en la naturaleza de ésta. Cuando la comunicación se realiza entre entidades a un mismo nivel empleamos el término protocolo, reservándose el de interfaz para los casos en que el enlace físico une entidades situadas en niveles distintos dentro de la jerarquía establecida anteriormente. Ejemplos de los primeros son el Data Link Control del American National Standards Institute [21], el Binary Synchronous Communication Protocol de I.B.M. [15] el High Level Data Link Control de la International Organization for Standarization [21, 23, 24], el Advanced Data Communication Control Procedure del ANSI [1], el Synchronous Data Link Control de I.B.M. [13] y el Digital

Data Communications Message Protocol de la Digital Equipment Corporation [10]. Como interfaces de este nivel podemos citar las recomendaciones X.20, X.21 y X.25 de la CCITT en lo referente al nivel 2 para la comunicación entre Equipo Terminal de Datos (ETD) y Equipo Terminal del Circuito de Datos (ETCD).

La variedad de las configuraciones de la conexión sobre el enlace físico (conexión multipunto o en lazo), de los modos de comunicación (half-duplex, full-duplex) de las técnicas de acceso a la misma (contención, roll call polling, hub polling) y la diversidad de los procedimientos para la recuperación de errores, sincronización, control de flujo, etc., empleados por los protocolos e interfaces de este nivel hace difícil el tratamiento uniforme de su estudio. Pueden resaltarse, sin embargo, las características comunes de todos ellos que derivan de la necesidad de realizar una serie de funciones genéricas de control del enlace.

La figura 2.10 muestra algunas configuraciones posibles de conexión. En el caso de conexiones punto a punto se distinguen dos formas de operación. La primera de ellas, denominada amo/esclavo (master/slave) supone el hecho de que una de las estaciones asume la responsabilidad del control de la comunicación, siendo necesaria la autorización específica de esta estación primaria para que la otra estación pueda transmitir información. Más frecuente es el caso en que ambas estaciones compiten por el uso de la línea bajo un principio de igualdad jugando cada una de ellas el papel de estación primaria en un sentido de la comunicación y de secundaria en el otro.

En las configuraciones multipunto y en lazo lo más frecuente es encontrar una estación primaria que ejerce el

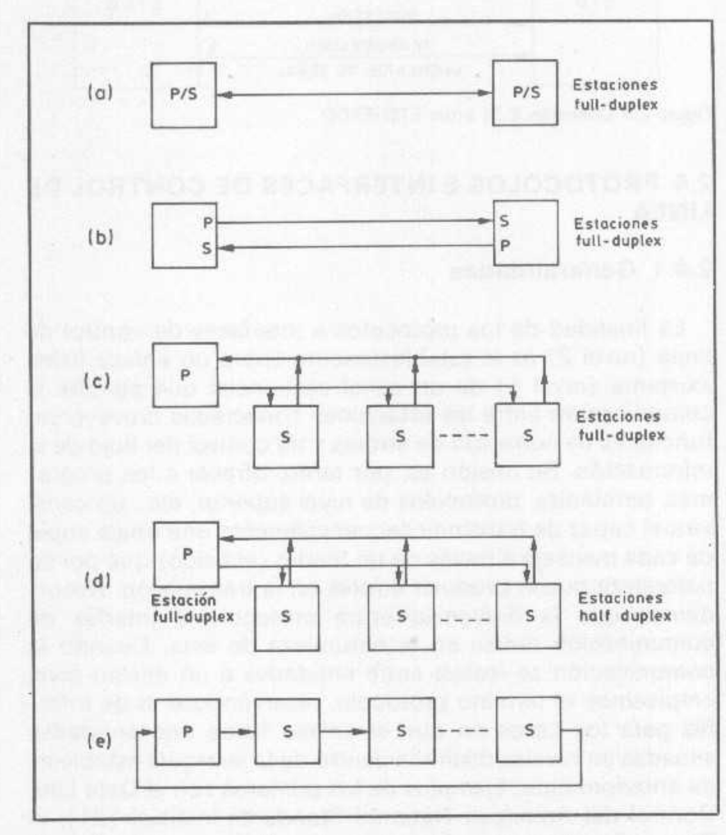


Figura 2.10 Algunas configuraciones posibles de la conexión. a) conexión punto a punto; modo de operación half-duplex; b) conexión punto a punto; modo de operación full-duplex; c) conexión multipunto; modo de operación full-duplex; d) conexión multipunto; modo de operación multimultipunto (P transmite y recibe simultáneamente operando con dos estaciones diferentes); e) conexión en un lazo: modo de operación simplex.

control del enlace físico, siendo todas las demás secundarias. La necesidad de mecanismos de direccionamiento y de control del acceso al enlace aparecen evidentes en este tipo de conexiones.

Las funciones de control del enlace son diferentes en las tres fases típicas en que puede encontrarse la comunicación: establecimiento del enlace, fase de transmisión y desconexión del enlace. Antes de comenzar el intercambio de información entre dos estaciones es necesario un preámbulo que conduzca a un estado en el que ambas estaciones queden en disposición de comenzar el coloquio. Esta fase conocida con el nombre de establecimiento del enlace, comporta el intercambio de bloques de control destinados a seleccionar y comprobar la disposición de la estación secundaria. En líneas generales, esta fase consta del envío por parte de la estación primaria de un bloque solicitando la atención de una estación secundaria; ésta contesta con un bloque indicando su aceptación o no aceptación de establecer una conexión dependiendo de su estado. Una vez establecida la conexión entre dos estaciones comienza la fase de transferencia de datos, durante la cual ambas entidades intercambian bloques conteniendo la información objeto de la conexión y realizan funciones de sincronismo y de control de los posibles errores.

Concluida esta fase es necesario terminar explícitamente la conexión dejando libre la línea para el establecimiento futuro de un nuevo enlace entre la estación primaria y una estación secundaria.

En el apartado anterior ha aparecido explícitamente el concepto de bloque, unidad de información empleada en las distintas fases de la conversación. Los protocolos e interfaces del nivel que estamos considerando habrán de especificar claramente los posibles formatos de los bloques de control de datos. A este respecto es necesario delimitar los campos de información de los bloques, especificar el o los alfabetos empleados en la codificación de la información, etc.

Con el fin de que la estación receptora de un bloque sea capaz de extraer e interpretar fielmente la información que contiene, es indispensable poder determinar el comienzo y final de cada octeto y de cada bit. Resulta necesario, por tanto, proveer al receptor con mecanismos capaces de establecer el sincronismo a nivel de bit y de octeto. En este sentido pueden indentificarse dos tipos de protocolos, según sea el bit o el carácter el elemento de información básico en la composición de los bloques, que reciben, respectivamente, el nombre de protocolos (o interfaces) orientados al bit o al carácter.

Un aspecto fundamental de los procedimientos de control de línea es la forma de conseguir la transparencia del texto. Se entiende por texto el conjunto de datos cuya transmisión de una estación a otra constituye el objeto de la comunicación. Resulta evidente que todo protocolo o interfaz ha de permitir la transferencia de cualquier combinación de "ceros" y "unos". La utilización de caracteres de control determinados en los procedimientos orientados al carácter, o de combinaciones especiales de bits para delimitar el inicio y fin de un bloque en el caso de los protocolos orientados al bit, hace necesario establecer mecanismos que permitan al receptor conocer cuando es parte de un texto una determinada combinación de bits y cuando se trata del código de una función de control.

La recuperación de los errores producidos en la transmisión sobre el enlace físico es otra faceta que han de contemplar los procedimientos de control de línea. El método más extendido consiste en incluir en los bloques

información que permita comprobar a la estación receptora su corrección. Esta información varía desde el conocido bit de paridad en las transmisiones del tipo start/stop al más completo CRC (cyclic redundancy check), pasando por los octetos de paridad vertical. En cualquier caso, la estación transmisora guarda una copia de cada bloque enviado hasta recibir la notificación explícita de recepción correcta de la estación destino.

Para la recuperación de errores detectados los protocolos e interfaces de comunicación de este nivel emplean frecuentemente la técnica PAR (Positive Acknowledgment-Retransmission on time-out) o una simplificación de ella [20, 35]. A la recepción de un bloque correcto la estación destino replica con una aceptación. La estación originaria del bloque de datos destruye la copia del mensaje cuando recibe esta aceptación. Si en un tiempo establecido (time-out) la estación generadora del bloque de datos no recibe la aceptación de la estación destino, debe suponer que la transmisión ha sido errónea, procediendo automáticamente a la retransmisión. Una técnica ampliamente utilizada para acelerar la retransmisión es enviar como respuesta a una recepción incorrecta un bloque de control indicando la no aceptación del bloque. A la recepción de este mensaje la estación origen retransmite el bloque no aceptado sin necesidad de esperar el fin de la temporización, pudiéndose incluso prescindir de ésta.

Esta forma de recuperación de los errores obliga a la estación transmisora a esperar la aceptación para poder enviar el bloque siguiente. Ello supone una pérdida en la eficiencia de la línea, particularmente notable si las estaciones operan en full-duplex. Con el fin de aprovechar mejor el enlace resulta conveniente poder transmitir más de un bloque sin tener que esperar la aceptación de uno para proceder al envío de otro. Algunos procedimientos de control de línea adoptan el denominado esquema de ventana para solventar este problema. Esta técnica requiere que los bloques lleven un número de secuencia que los identifique. Análogamente, la aceptación lleva un número que sirve de aceptación a un bloque en particular o a todos los bloques enviados con número de secuencia inferior (o igual).

La numeración de los bloques es cíclica y no pueden enviarse más de una cantidad determinada de bloques sin recibir la aceptación del primero de ellos. Esta cantidad recibe el nombre de ventana de transmisión.

A continuación se resumen las características generales de algunos de los protocolos e interfaces de comunicación de este nivel más representativos, no pretendiéndose en modo alguno ofrecer una especificación formal y completa de ellos.

2.4.2 ANSI DLC, IBM BSC

El protocolo DLC, compatible con el ISO R646, fue creado para sustituir un conjunto disperso de protocolos orientados al carácter diseñados para sistemas muy variados. Este protocolo es similar al estándar de IBM que recibe el nombre de BSC. Las fases fundamentales de su operación son el establecimiento de la conexión, la transferencia de datos y la desconexión. Cada una de estas fases comporta el intercambio de bloques formados por caracteres. Las diferentes funciones y campos de los bloques que se transmiten se especifican mediante un conjunto de caracteres del código ASCII (American Standard Code for Information Interchange), lo que permite una longitud variable de los diversos campos.

2.4.2.1 *Caracteres de control de la comunicación*

SYN (SYNchronous idle). Todo bloque en las transmisiones síncronas va precedido de uno o dos caracteres SYN. La estación receptora atiende continuamente a la aparición en el enlace físico de la combinación SYN(SYN) para identificar el comienzo del bloque. Este mecanismo constituye la base del sincronismo de bit y de carácter, es decir, le sirve al receptor para saber dónde comienza y acaba cada bit y cada carácter del resto del bloque. En el caso de que la longitud de éste sea grande, el transmisor deberá incluir periódicamente en el bloque caracteres SYN(SYN) para facilitar la resincronización del receptor.

SOH (Start Of Heading). Indica que los caracteres que le siguen constituyen la denominada cabecera del bloque. La cabecera hace referencia exclusivamente al bloque al que pertenece, y contiene información indicando la dirección de la estación destino, de la ruta que ha de seguir el bloque, y tiene funciones de control en general. La cabecera se termina con otro carácter de control (STX o ENQ).

STX (Start of TeXt). La información objeto de la transmisión, el texto, se configura como una secuencia de caracteres que puede viajar en uno o más bloques. Los bloques que contienen texto indican su comienzo precediéndole del carácter de control STX.

ETB (End of Transmission Block). Este carácter de control sirve para indicar el final de un bloque que contiene texto (o una cabecera), señalando que no es final del texto, es decir, que han de venir más bloques conteniendo texto.

ETX (End of TeXt). El bloque que contiene el último trozo de un texto enviado en múltiples bloques termina con un caracter ETX en vez del ETB.

ACK (ACKnowledge). Cuando la estación receptora considera que un bloque recibido no contiene errores notifica su aceptación a la estación transmisora mediante este carácter. Su significado en la fase de establecimiento del enlace es la disposición de la estación secundaria a la realización de la conexión. En ocasiones puede llevar una numeración secuencial módulo 2, es decir, ACK0 o ACK1.

NAK (Not AcKnowledge). Se emplea para notificar a la estación transmisora la recepción incorrecta de un bloque o la imposibilidad de establecer el enlace.

ENQ (ENQuiry). Este carácter sirve para exigir a la estación secundaria una respuesta. Se emplea para identificar a la estación receptora o para solicitar información sobre su estado. La transmisión de este carácter por la estación primaria inicia la fase de establecimiento del enlace.

EOT (End Of Transmission). Este carácter concluye la transmisión, iniciando la fase de desconexión.

DLE (Data Link Escape). Su misión es cambiar el significado de un número determinado de los caracteres que le siguen en el bloque. Puede emplearse por tanto para definir funciones de control específicas del usuario. La transparencia del texto se consigue en el BSC mediante este carácter. En efecto, para evitar que la aparición dentro de un texto o cabecera de un carácter de información cuyo código coincida con uno de los nueve caracteres de control más arriba definidos sea interpretado erróneamente por el receptor, confundiéndolo con el carácter de control correspondiente, se antepone a los caracteres de control un DLE. Resta resolver el problema de conseguir la transparencia en los datos del carácter DLE. Para ello la estación transmisora duplica todo carácter del texto o cabecera cuyo código sea el del DLE, siendo misión del receptor la eliminación de este carácter adicional.

Además de estos diez caracteres de control el BSC posee

algunos más (ITB, end of Intermediate Text or heading Block; WACK, ACKnowledge but Wait; etc.).

Cada bloque de datos transmitido se termina con un carácter obtenido haciendo la "O exclusiva" de todos los caracteres del bloque a partir del primer SOH o STX. Este carácter recibe el nombre de BCC (Block Chek Character).

El estándar de ANSI es en realidad un conjunto de reglas cuya aplicación difiere según el tipo de comunicación que se desee establecer y las características de las estaciones, dejando además la posibilidad de que cada usuario defina reglas propias adicionales con el fin de configurar el protocolo a la medida de sus necesidades. Los tipos de enlace a que puede aplicarse este estándar son, entre otros:

a) Transmisión en un solo sentido desde una estación primaria a varias secundarias conectadas a un enlace multipunto.
b) Transmisión alternante en dos sentidos entre dos estaciones conectadas en configuración punto a punto.
c) Transmisión alternante en dos sentidos entre una estación primaria y varias secundarias conectadas a un enlace multipunto.

El protocolo BSC se diseñó exclusivamente para los dos últimos tipos de comunicación señalados para el ANSI DLC.

A continuación se esquematizan algunos ejemplos de posibles conversaciones prescindiendo de las diferencias entre ambos protocolos y de ciertos caracteres de control como los SYN y DLE. Su objeto es facilitar la comprensión del funcionamiento de este tipo de protocolos en las distintas fases de establecimiento del enlace, transmisión de datos y desconexión.

La figura 2.11*a* considera una configuración del tipo (a), señalado más arriba. El establecimiento de la conexión se realiza entre la estación primaria y una secundaria mediante el envío de un carácter ENQ precedido por una secuencia que incluye la dirección de la estación secundaria y otros caracteres posiblemente definidos por el usuario. La estación secundaria direccionada debe sincronizarse con este bloque. Cada vez que la estación primaria desea cambiar el destino de sus bloques ha de acabar la conexión que mantenía anteriormente. La figura 2.11*b* muestra una secuencia de bloques emitida por la estación primaria.

La figura 2.12*a* representa dos estaciones que se comunican bloques alternativamente sobre un enlace half-duplex. En esta configuración cada estación es primaria respecto a la transmisión y secundaria para la recepción. Ambas pueden intentar el establecimiento del enlace enviando un ENQ y cerrarlo mediante un EOT. La estación que recibe un ENQ ha de responder con un bloque que indique su disposición ante la petición de realizar la conexión. En la conversación de la figura 2.12*b* la estación 1 envía un texto en dos bloques a la 2 y posteriormente un segundo texto de un único bloque. A continuación la estación 2 envía a la 1 un bloque que contiene un texto. Obsérvese que la estación que hace de secundaria responde con un NAK a la primera petición de apertura del enlace, solicitando así la retransmisión del bloque.

Para concluir considérese la conexión centralizada multipunto de la figura 2.13*a*. En este contexto la estación primaria puede seleccionar una secundaria como destino de la información a transmitir o, por el contrario, designarla temporalmente como primaria para que transmita información. La figura 2.13*b* ilustra estos conceptos. Primeramente la estación primaria envía a S1 un texto de un bloque, solicitando a continuación información de S2. Ésta envía un texto en 2 bloques, cerrando ambas la conexión. Posteriormente la estación primaria designa a S1 para que transmita información, lo que hace S1 envíando un bloque de texto e indicado con un EOT que no tiene más información para enviar. A continuación P envía a S1 un bloque de información, cerrando después la conexión.

Las limitaciones de los protocolos ANSI DLC e IBM BSC son múltiples. Entre ellas cabe destacar:

— no son verdaderos estándares;

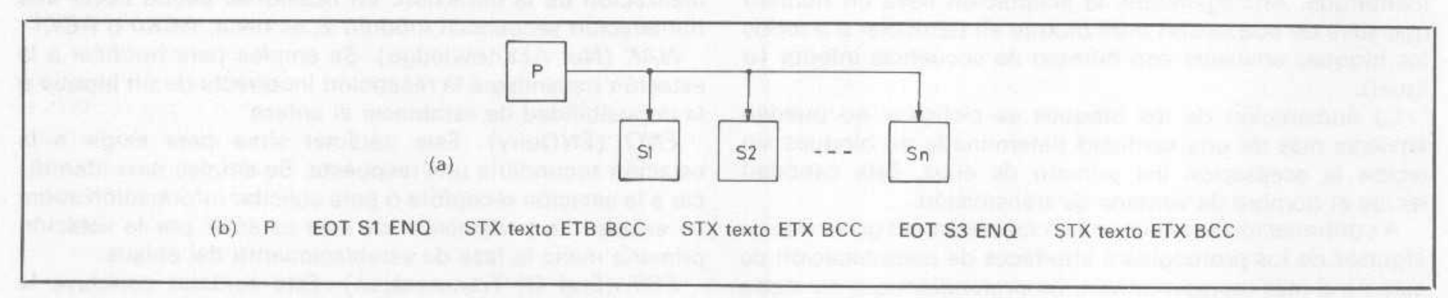


Figura 2.11 (a) Conexión multipunto con transmisión en un único sentido; (b) Posible conversación

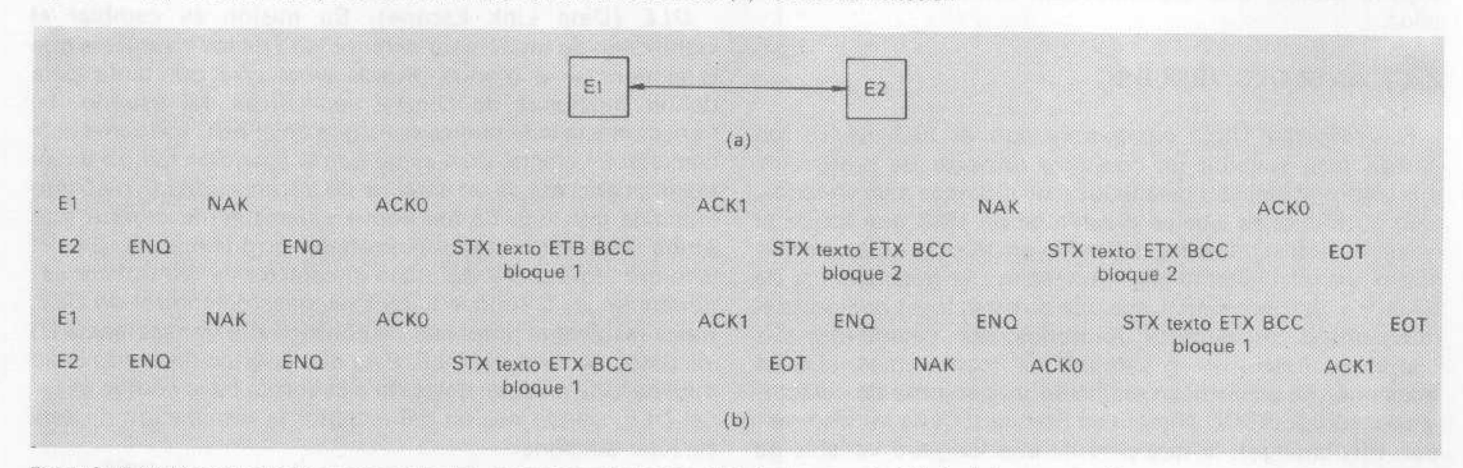


Figura 2.12 (a) Conexión punto a punto entre dos estaciones que alternan el envío de mensajes; (b) Posible conversación

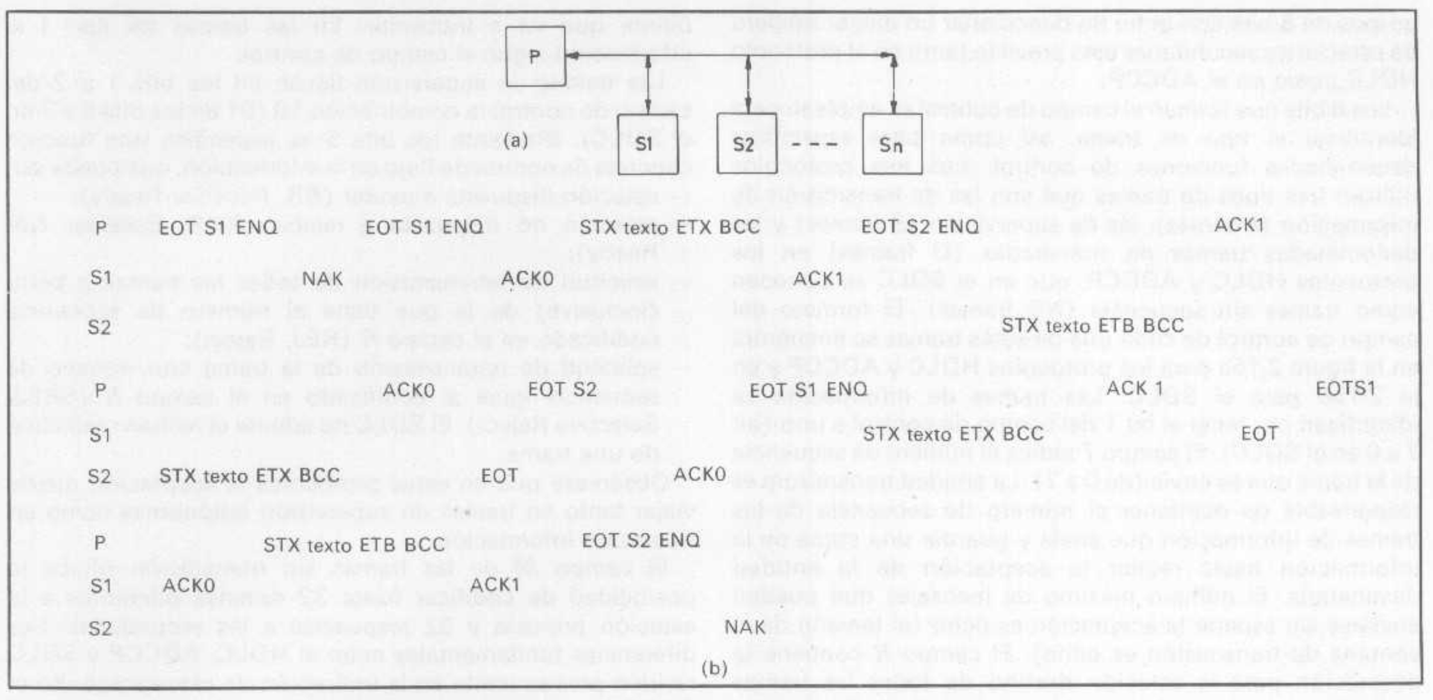


Figura 2.13 (a) Conexión multipunto entre una estación primaria y varias secundarias; (b) Posible conversación

— no son detectables los errores que se producen en los bloques de control;
— sólo pueden utilizar enlaces half-duplex (y simplex el DLC);
— no permiten configuraciones en lazo;
— la forma de conseguir la transparencia es ineficiente;
— no permiten el envío de un nuevo bloque hasta la aceptación del anterior, degradando así el rendimiento del enlace;
— el procedimiento de recuperación de errores impide la utilización de la línea al resto de las estaciones.

Para solventar estos problemas y ofrecer un marco más amplio de formas de operación se han diseñado diversos protocolos de control de línea llamados a sustituir totalmente al DLC y BSC. Dejamos este tema para el siguiente subapartado.

2.4.3 ISO HDLC, ANSI ADCCP, IBM SDLC

Para la creación de canales virtuales entre elementos adyacentes de un sistema distribuido capaces de transmitir información de forma transparente y sin errores, diversas organizaciones han propuesto protocolos orientados al bit cuya similitud es notable. Entre ellos destacan el HDLC (ISO), el ADCCP (ANSI) y el SDLC (IBM). En este apartado se estudian los aspectos comunes de estos tres protocolos, haciéndose referencia explícita a algunas particularidades de cada uno de ellos.

Varias son las formas de conexión con las que que estos protocolos han de ser capaces de trabajar, fundamentalmente conexiones sobre enlaces punto a punto, sobre enlaces multipunto y conexiones en lazo. En cualquier caso el protocolo ha de ser capaz de establecer la comunicación entre dos entidades, lo que en el caso de la conexión multipunto y en lazo exige la identificación de una estación emisora y una o varias receptoras. Los tres protocolos que estamos considerando funcionan bajo el principio de designar una estación de las conectadas al enlace físico como primaria. Es responsabilidad de la estación primaria la iniciación y autorización de la transmisión al resto de estaciones secundarias. En el caso de la comunicación punto a punto cada una de las dos estaciones es estación primaria en un sentido de la transmisión y secundaria en el opuesto.

Según si el enlace físico es único o doble el protocolo es capaz de funcionar en distintos modos de operación. Por ejemplo, el SDLC puede trabajar sobre un único enlace en modo half-duplex o simplex (en el caso de una configuración en lazo) y sobre un enlace doble en modo full-duplex-multimultipunto para estaciones secundarias que trabajan en half-duplex, o full-duplex para estaciones secundarias que operan en full-duplex.

Los tres protocolos adoptan una forma similar en la constitución de los bloques de transmisión que reciben el nombre de tramas. Independientemente del tipo de mensaje de que se trate, cada trama está compuesta de un conjunto de campos fijos tal como muestra la figura 2.14. El campo *F* que inicia y concluye cada trama es un octeto cuyos bits son 01111110. El receptor busca continuamente en la línea esta agrupación de bits y una vez localizada le sirve para sincronizarse con la trama que le llega. El final de la trama se detecta de forma análoga. Para conseguir la transparencia de la información dentro de la trama es necesario inhibir la posibilidad de la secuencia *F* en la misma; de otro modo sería interpretada por el receptor como el fin de la trama. Por ello el transmisor inserta un bit 0 tras la aparición de cualquier secuencia de cinco bits 1, impidiendo la formación de secuencias de seis unos. Análogamente el receptor suprime de la información todo bit a 0 que aparezca a continuación de una secuencia de cinco unos.

El campo de dirección de cada trama identifica la estación destino entre un total de 256. La extensión de este campo en



Figura 2.14 Formato de la trama de los protocolos orientados al bit

grupos de 8 bits con el fin de direccionar un mayor número de estaciones secundarias está previsto tanto en el protocolo HDLC como en el ADCCP.

Los 8 bits que forman el campo de control se emplean para identificar el tipo de trama, así como para especificar determinadas funciones de control. Los tres protocolos utilizan tres tipos de tramas que son las de transmisión de información (I frames), las de supervisión (S frames) y las denominadas tramas no numeradas (U frames) en los protocolos HDLC y ADCCP, que en el SDLC se conocen como tramas sin secuencia (NS frames). El formato del campo de control de cada una de estas tramas se encuentra en la figura 2.15*a* para los protocolos HDLC y ADCCP y en la 2.15*b* para el SDLC. Las tramas de información se identifican por tener el bit 1 del campo de control a uno (bit 7 a 0 en el SDLC). El campo *T* indica el número de secuencia de la trama que se envía (de 0 a 7). La entidad transmisora es responsable de mantener el número de secuencia de las tramas de información que envía y guardar una copia de la información hasta recibir la aceptación de la entidad destinataria. El número máximo de mensajes que pueden enviarse sin esperar la aceptación es ocho (el tamaño de la ventana de transmisión es ocho). El campo *R* contiene la aceptación para la estación destino de todas las tramas recibidas de ella en la estación que envía la trama con número de secuencia inferior al que se codifica en *R*, siendo por tanto este número el que se espera en la próxima trama. Obviamente, en una conexión multipunto el transmisor tendrá que tratar independientemente las aceptaciones de las tramas que recibe de cada estación secundaria. El bit P/F (Poll/Final) tiene un significado diferente según sea el emisor de la trama una estación primaria o secundaria. La estación primaria exige, mediante este bit, a la secundaria una trama de respuesta. Mediante el bit P/F las estaciones secundarias pueden notificar a la primaria que la trama es la

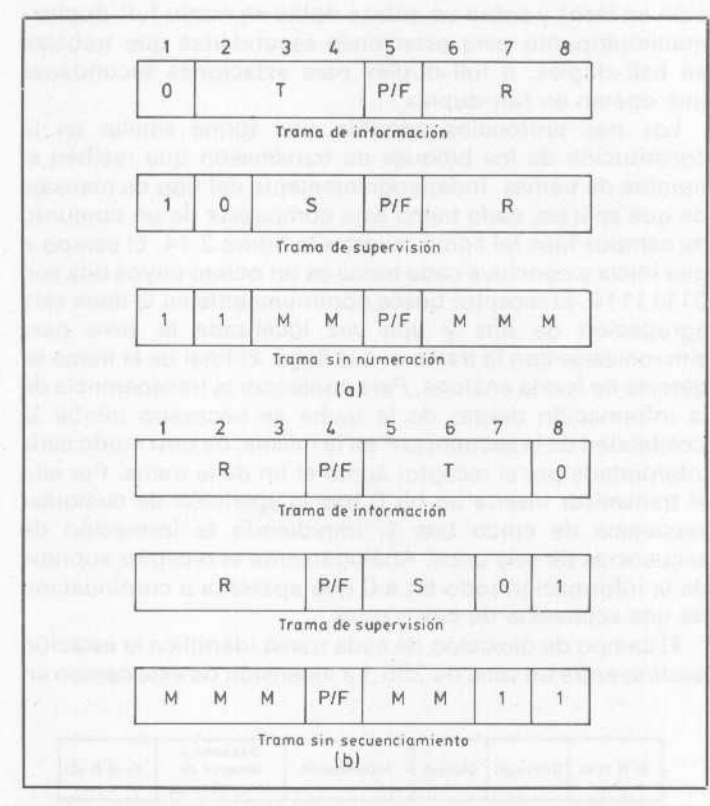


Figura 2.15 (a) Formato del campo de control del HDLC y ADCCP; (b) Formato del campo de control del SDLC

última que va a transmitir. En las tramas de tipo I la información sigue al campo de control.

Las tramas de supervisión llevan en los bits 1 y 2 del campo de control la combinación 10 (01 en los bits 6 y 7 en el SDLC). Mediante los bits S se especifica una función concreta de control de flujo de la información, que puede ser:

- estación dispuesta a recibir (RR, Receiver Ready);
- estación no dispuesta a recibir (RNR, Receiver Not Ready);
- solicitud de retransmisión de todas las tramas a partir (inclusive) de la que tiene el número de secuencia codificado en el campo *R* (REJ, Reject);
- solicitud de retransmisión de la trama con número de secuencia igual al codificado en el campo *R* (SREJ, Selective Reject). El SDLC no admite el rechazo selectivo de una trama.

Obsérvese que en estos protocolos la aceptación puede viajar tanto en tramas de supervisión autónomas como en tramas de información.

El campo *M* de las tramas sin numeración ofrece la posibilidad de codificar hasta 32 órdenes diferentes a la estación primaria y 32 respuestas a las secundarias. Las diferencias fundamentales entre el HDLC, ADCCP y SDLC residen precisamente en la utilización de este campo. En el ADCCP las órdenes hacen referencia al modo de operación, conclusión de la conexión, identificación de las estaciones, etc. El SDLC utiliza también las tramas NS para transmitir información sin número de secuencia.

Los modos de operación concretan la actividad transmisora de la estación secundaria. En el modo NRM (Normal Response Mode) la estación secundaria sólo puede iniciar la transmisión a petición de la primaria. Por el contrario, trabajando en ARM (Asynchronous Response Mode) una estación secundaria puede comenzar la transmisión sin esperar una orden de la primaria.

En los tres protocolos la determinación en la recepción de los errores de transmisión se realiza atendiendo al valor del campo de 16 bits que precede al fin de la trama. El contenido de este campo se obtiene mediante el polinomio generador $x^{16}+x^{12}+x^5+1$ (recomendación V.41 CCITT), afectando a los campos de dirección, control e información.

2.4.4 CCITT X.20, X.21, X.25 en el nivel 2

Recordemos que las interfaces X.20 y X.21 son recomendaciones CCITT para comunicar ETD y ETCD sobre enlaces multipunto y punto a punto, estando la primera orientada al funcionamiento arrítmico y la segunda al síncrono. Al nivel 2 especifican las posibles formas de sincronización y parte del procedimiento para establecer el enlace.

La sincronización de carácter en la X.20 se realiza por el conocido método de marcha/paro (start/stop), en el cual cada octeto va precedido de un bit que marca el comienzo del carácter, indicándose su fin mediante dos bits. La X.21 establece la sincronización de carácter mediante dos mecanismos alternativos. El primero consiste en el empleo de dos caracteres SYN precediendo toda secuencia de caracteres. El segundo hace referencia a las entidades que disponen del circuito de muestreo de carácter definido en X.24. Este circuito suministra un pulso de sincronismo de carácter a DTE que le permite sincronizar la información en los circuitos *R* y *T*. El control de errores emplea un único bit de paridad impar.

La recomendación X.25 [3, 6] especifica la interfaz ETD y ETCD para terminales conectados a una red pública de conmutación de paquetes. Es en realidad una interfaz entre

la red de comunicaciones (nivel 3) y sus usuarios (nivel 4). Sin embargo, el X.25 especifica los protocolos a utilizar en los niveles inferiores. Para el nivel 1 se emplea la norma X.21 y para el nivel 2 se define un protocolo denominado LAP (Link Access Protocol), similar al HDLC sin ser completamente compatible. Una especificación posterior del LAP, el LAP B, resulta ser totalmente compatible con éste.

2.4.5 DEC DDCMP

Este protocolo fue diseñado por DEC, orientándolo a resolver los problemas de comunicación entre computadores. Según sus especificaciones [10], el DDCMP define la estructura, contenido y procedimientos de secuenciamiento para la transmisión de información entre computadores, así como las técnicas empleadas para la detección y recuperación de errores. La función del protocolo es el intercambio de mensajes, asegurando el secuenciamiento correcto y la integridad de la información sobre enlaces físicos de comunicación que tienen una probabilidad considerable de introducir errores. En el diseño se han tomado en cuenta los problemas de la inicialización de un computador por otro (bootstrap). El protocolo es simétrico y permite la operación half y full-duplex tanto en conexiones punto a punto como multipunto.

El DDCMP está orientado al carácter y los tipos de mensaje que define —DEC no emplea el término trama— son:

a) mensajes numerados (para la transmisión de datos);
b) mensajes no numerados (para funciones de control).

La figura 2.16 muestra el formato de los mensajes numerados y la figura 2.17 el de los no numerados, así como una lista de los caracteres de control junto a una breve indicación de su utilización.

Es interesante observar que en este protocolo la longitud del campo de datos se especifica mediante dos caracteres de la cabecera. Para la detección de errores se emplea el mismo polinomio generador que el HDLC, etc., llevando los mensajes de datos dos campos de detección, uno para la cabecera y otro para los datos. El protocolo es de tipo PAR, por lo que los mensajes se retransmiten a la recepción de un NAK o al transcurrir un tiempo predeterminado desde el envío del mensaje en caso de no recibir un ACK o NAK. El mensaje de control NAK puede originarse, además de como réplica a un mensaje erróneo, en el caso de que el

SS	Secuencia de sincronización
Soh	inicio de cabecera
CC1	número de caracteres entre bcc2 y bcc3 exclusive
CC2	codificado en 14 bits (6 de CC2 y 8 de CC1)
final	los flags final y select se emplean para controlar la adquisición del
Select	enlace
r	número de secuencia del último mensaje recibido correctamente; el mensaje numerado con r y todos los anteriores se han recibido correctamente en la estación que transmite este mensaje
n	número de secuencia del mensaje
a	dirección de la estación de destino
bcc1	bloque de control de errores obtenido mediante el polinomio
bcc2	generador $X^{16}+X^{15}+X^2+1$ desde *soh* hasta *a*
data1	caracteres de datos en número igual al definido
data 2	por CC1 y CC2
⋮	
bcc3	bloque de control sobre los caracteres de datos obtenido mediante
bcc4	el polinomio indicado más arriba.

Figura 2.16 Formato de los mensajes numerados del DCCMP

enq	enquiry (interrogación)
type	tipo de mensaje no numerado
sub-type	subtipo, su significado depende del tipo
final	los flags final y select se emplean para controlar la adquisición del
select	enlace
r	campo de respuesta; su significado depende del tipo
n	campo de numeración; su significado depende del tipo
a	dirección de la estación de destino
bcc1	bloque de control sobre los caracteres del mensaje obtenido
bcc2	mediante el polinomio $X^{15}+X^{12}+X^2+1$.

(a)

ack	(acknowledge). Aceptación.
nak	(not acknowledge). Rechazo
rep	(reply to message number). Se envía de una estación primaria a una secundaria cuando durante un tiempo predeterminado la primaria no ha recibido ningún mensaje de la secundaria y tiene mensajes pendientes de aceptación. La respuesta esperada es un ack ó nak
res	(reset message number). Enviado por una estación primaria a una secundaria obligando a que ésta ponga a 0 su número de secuencia de mensajes recibidos.
resak	(reset acknowledge). Respuesta a res.
strt	(start). Enviado por una estación primaria a una secundaria para comenzar el intercambio de mensajes
stack	(start acknowledge). Respuesta a strt.
boot	(bootstrap). La misión de este mensaje es transferir un programa a la estación receptora.

(b)

Figura 2.17 (a) Formato de los mensajes no numerados del DCCMP; (b) Tipo de los mensajes no numerados

computador que recibe el mensaje no disponga de memoria para almacenarlo, o bien por la recepción de un mensaje que no comienza por SOH, ENQ o DLE (empleado para sincronización de modems asíncronos).

Como en otros protocolos, el ACK puede enviarse en un mensaje de control específico o sobre los mensajes de datos. En este caso, el campo <r> indica el número del último mensaje que se ha recibido correctamente, significando la aceptación de este mensaje y de todos los anteriores con numeración inferior. El número de mensajes que pueden enviarse sin esperar la aceptación del primero es 256. Cuando una estación recibe un mensaje cuyo número de secuencia es distinto del esperado no produce respuesta alguna, ignorándose, además, la información contenida en éste.

Los campos <select> y <final> sirven para controlar el acceso de las estaciones al enlace, dependiendo su significado exacto de la forma de operación.

El protocolo establece la siguiente prioridad para la transmisión de los mensajes:

— BOOTstrap (BOOT)
— STart ACKnowledge (STACK)
— STaRT (STRT)
— RESet (RES)
— Negative AcKnowledge (NAK)
— REPLy (REP)
— Numbered Messages (NM)
— ACKnowledge (ACK)

Obsérvese que la entidad que haya de realizar el protocolo debe disponer de memoria suficiente para el almacenamiento de un número relativamente grande de mensajes, así como de mecanismos adecuados para medir el tiempo transcurrido entre emisiones y recepciones. Esto no supone en general problema alguno, puesto que las entidades llamadas a realizar este protocolo son, en principio, computadores.

2.5 PROTOCOLOS E INTERFACES DE CONTROL DE LA RED

2.5.1 Introducción

Hemos visto cómo los protocolos de control de línea crean un canal de comunicaciones libre de errores de transmisión entre entidades conectadas a través de un enlace físico. En muchas ocasiones es necesario establecer conexiones entre entidades que se encuentran separadas por más de un enlace físico. Considérese, por ejemplo, el caso de la transmisión de información de un nudo de una red de comunicaciones a otro nudo al que no está unido por un enlace. Para controlar el flujo de información en estos casos es necesario disponer de protocolos específicos [4, 7, 11, 32, 35, 40, 43].

Es misión de los nudos de la red determinar la ruta que ha de seguir un mensaje que le llega con destino diferente de alguno de los usuarios a los que está físicamente conectado. La determinación de la ruta se basa en la dirección del nudo destino y en la aplicación de algoritmos de ruta apropiados. En cualquier caso, el software de los nudos de comunicación, junto con los canales virtuales creados por los protocolos de control de línea, han de ser capaces de transportar un mensaje desde el nudo origen de la red al nudo destino.

El cometido de los protocolos del subnivel de transporte es precisamente crear un canal virtual que permita la transferencia de información entre el nudo origen y destino. Algunos protocolos de este nivel, como son los protocolos de transporte de la red Cyclades o de DECNET, no poseen funciones de detección y recuperación de errores ni de control de flujo. Sin embargo, lo más frecuente es encontrar en este nivel mecanismos similares a los de los protocolos de control de línea que permiten crear un canal entre los nudos origen y destino que garantice la transmisión correcta de los mensajes, es decir, en la secuencia adecuada, sin duplicaciones ni errores.

Un esquema ampliamente utilizado en el nivel de transporte es el esquema de ventana en trasmisión y recepción. En cada sentido de la comunicación el nudo emisor dispone de una ventana de transmisión cuyo límite inferior es el número de secuencia del primer mensaje enviado del que aún no se ha recibido la aceptación. El límite superior de la ventana es el límite inferior más una constante del sistema, que no es otra cosa que el número de mensajes que pueden enviarse sin recibir aceptación. Por su parte, el receptor mantiene una ventana de recepción cuyo límite inferior es el número de secuencia inmediatamente superior al del último mensaje recibido correctamente y pasado al usuario. Dado que en su viaje a través de la red de los mensajes pueden "desordenarse", llegando al nudo destino en orden diferente del que fueron transmitidos por el nudo emisor, el receptor acepta mensajes correctos cuyo número de secuencia sea inferior al límite de la ventana de recepción. Estos mensajes no puede pasarlos al usuario, sino que debe almacenarlos temporalmente. En otras palabras, la entidad receptora acepta y almacena los mensajes recibidos sin error con número de secuencia comprendido entre el límite inferior de la ventana de repeción y este valor más una constante del sistema.

Los mecanismos de control y recuperación de errores son también similares a los de los protocolos de control de línea, siendo los más extendidos los protocolos de tipo PAR. El nudo emisor guarda una copia del mensaje enviado y tras la emisión pone en marcha un temporizador. El receptor responde con un mensaje de aceptación o no aceptación según la recepción haya sido o no correcta. La recepción del mensaje de aceptación en el nudo transmisor hace que la copia del mensaje se destruya, mientras que la recepción de la no aceptación o el fin de la temporización provocan la retransmisión del mensaje.

El subnivel de transporte garantiza al nivel superior que lo utiliza la entrega correcta de la información en el destino que se le indique, supuesto que este subnivel sea responsable del control de errores y de flujo. Es tarea de los protocolos fin a fin el establecimiento de un circuito virtual entre dos entidades antes de comenzar el envío de la información. En general, las fases de conexión a este nivel son el establecimiento del circuito, la fase de transferencia de información y la desconexión del circuito virtual. Se comprende fácilmente que un par de computadores conectados respectivamente a dos nudos de la red de comunicaciones pueden desear establecer varias conversaciones simultáneamente. Los protocolos del subnivel fin a fin habrán de ser capaces de gestionar el establecimiento de varios circuitos virtuales que operen al mismo tiempo. Cuando un proceso de nivel superior al nivel de control de la red desea comunicarse con otro proceso localizado en otro computador debe solicitar el establecimiento de un circuito virtual. La solicitud se materializa, en general, en una llamada al software de comunicaciones que realiza el protocolo fin a fin. Éste habrá de iniciar la creación de un circuito virtual que, una vez establecido, permita la transferencia de datos entre los procesos.

Otra función característica de este nivel es la fragmentación de la información que se le entrega para transmitir en unidades más pequeñas o paquetes y la posterior reconstrucción en el nudo receptor de la información. En algunos sistemas distribuidos, como por ejemplo en la red Arpanet, se ha visto la conveniencia de no iniciar la transmisión de los paquetes hasta tener la garantía que en el nudo destino hay suficiente memoria para albergar todos los paquetes que integran el mensaje. De no hacerse la reserva de espacio se corre el riesgo de llegar a una situación en que el nudo destino tenga la memoria llena de mensajes parcialmente reconstruidos correspondientes a varias conversaciones. Como el mensaje no puede entregarse hasta estar completo y como no pueden aceptarse por falta de espacio los nuevos paquetes que completarían los mensajes, es ésta una situación de bloqueo. En algunos protocolos fin a fin pueden encontrarse, por tanto, mecanismos destinados a la reserva de espacio.

A continuación se exponen las características fundamentales de una interfaz de este nivel que pueden servir para comprender la problemática y las soluciones empleadas en el control de la red.

2.5.2 La recomendación X.25

X.25 es la interfaz de comunicación recomendada por el CCITT en la interconexión entre ETD y ETCD para equipos que funcionan en el modo paquetes en redes públicas de datos [18]. Para los niveles 1 y 2 se establecen las recomendaciones X.21 (bis) y el procedimiento de acceso al enlace (LAP). El nivel 3 de la recomendación especifica los procedimientos referentes a la utilización de circuitos virtuales permanentes y no permanentes. Estos últimos reciben el nombre de llamadas virtuales. La unidad de transmisión entre ETD y ETCD a este nivel se denomina paquete. Cada paquete se transfiere en el campo de información de una trama del protocolo de control de enlace.

En la realización de las llamadas virtuales y los circuitos virtuales permanentes se emplean canales lógicos. A cada llamada virtual o circuito permanente se le asigna un número entre 0 y 15 que identifica un grupo de canales lógicos y un número entre 0 y 255 que referencia un canal concreto dentro del grupo. En las llamadas virtuales se asigna el número de canal lógico durante la fase de establecimiento del circuito. Toda la información que circule por el canal virtual lleva el número de canal lógico que utiliza. Los circuitos permanentes ofrecen una conexión fin a fin por un período de tiempo indefinido, no siendo necesarios los procedimientos de establecimiento y liberación de la conexión, que siempre está establecida. El establecimiento y liberación de llamadas virtuales se realiza mediante el intercambio de paquetes de control. Las acciones pueden ser iniciadas por el ETD de la interfaz ETD/ETCD considerada o por otro ETD. Los números de canal lógico que no se están utilizando corresponden a canales lógicos que se encuentran en estado preparado. El ETD que quiere iniciar una llamada lo hace enviando un paquete de petición de llamada, seleccionando un canal lógico que pasa a estado "en espera". El ETCD destino de este paquete indica a su ETD que hay un paquete de llamada entrante, asociando a la llamada el canal lógico de número menor que está en estado preparado. El ETD llamado transferirá un paquete de llamada aceptada que especifique el mismo número de canal lógico que el paquete de llamada entrante. A la recepción de este paquete en el ETD que inicia la llamada, el canal lógico pasa al estado de transferencia de datos. La liberación del canal sigue un procedimiento similar.

Los paquetes de datos transmitidos en ambos sentidos de la interfaz ETD/ETCD se numeran módulo 8 con el número secuencial de paquete en emisión. La longitud máxima del campo de datos en estos paquetes es de 128 octetos (ampliable). La recuperación de errores en la fase de transferencia de datos adopta un esquema de ventana similar al expuesto en la introducción de este apartado, por lo que no se repite aquí. La recomendación X.25 define toda una serie de paquetes de control, así como las acciones a tomar en cada caso tanto por el ETD como por el ETCD. Este último aspecto se especifica mediante diagramas de estado que definen completamente los sucesos en la interfaz ETD/ETCD en cada canal lógico.

El primer octeto de cada paquete que se transmite lleva codificado en 4 bits el número de grupo de canales lógicos con el que el paquete ha de viajar. Los 4 bits restantes son el campo identificador de formato y su contenido indica el formato que se utiliza en el resto de la cabecera del paquete. El segundo octeto de cada paquete (excepto en un caso) contiene el número de canal lógico. El tipo de paquete se codifica en el tercer octeto, distinguiéndose entre los paquetes de establecimiento y reinicialización y, por último, de reanudación. La figura 2.18 ilustra a título de ejemplo, el formato de algunos paquetes X.25.

Esta recomendación es muy importante, dado que al haber sido aceptada tanto en Europa como en EE.UU. sienta las bases de la comunicación en las redes públicas de datos en modo paquetes. Actualmente se trabaja para eliminar los defectos técnicos de esta especificación, así como para extenderla con otras formas de comunicación (datagramas).

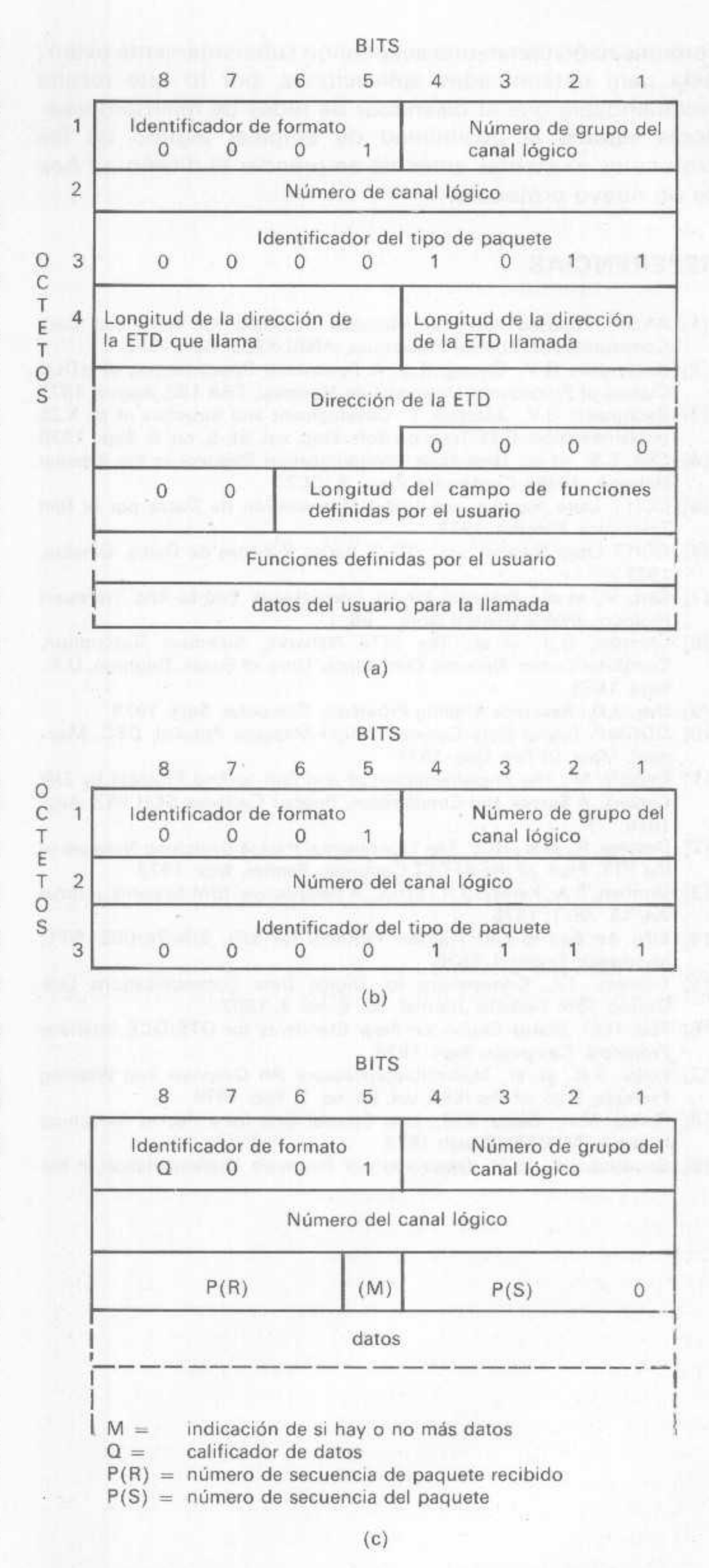


Figura 2.18 (a) Formato de los paquetes de petición de llamada y llamada entrante de X.25; (b) Formato de los paquetes de aceptación de llamada y llamada conectada de X.25; (c) Formato de los paquetes de datos de X.25

2.6 RESUMEN

Para la construcción de multimicroprocesadores y redes de microprocesadores es necesaria la comprensión de las formas posibles en que puede materializarse la cooperación entre procesos situados en procesadores distintos. En este capítulo se han expuesto las características fundamentales de los protocolos de comunicación que rigen el intercambio de información en las redes de computadores, limitando la exposición a aquellos aspectos que presumiblemente vaya a encontrar con más frecuencia el diseñador de redes de microprocesadores. Algunos estándares de protocolos de

comunicación tienen una aceptación suficientemente extendida para determinadas aplicaciones, por lo que resulta recomendable que el diseñador de redes de microprocesadores estudie la posibilidad de emplear alguno de los protocolos existentes antes de emprender el diseño *ad hoc* de un nuevo protocolo.

REFERENCIAS

[1] ANSI, Proposed American National Standard for Advanced Data Communication Control Procedures, ANSI X3S3, Dec. 1973
[2] Bochmann, G.V., Chung, R.J., A Formalized Specifications of HDLC Classes of Procedures, Universite de Montreal, TRA 152, August 1977
[3] Bochmann, G.V., Joachim, T., Development and Structure of an X.25 Implementation, IEEE Tran. on Soft. Eng., vol. SE-5, no. 5, Sept. 1979
[4] Carr, C.S., et al., Host-Host Communication Protocol in the Arpanet Network, AFIPS Conference Proc., SJCC70
[5] CCITT Libro Naranja, vol. VIII-1, Transmisión de Datos por la Red Telefónica, Ginebra, 1977
[6] CCITT Libro Naranja, vol. VIII-2, Redes Públicas de Datos, Ginebra, 1977
[7] Cerf, V., et al., Proposal for an Internetwork End-to-End Transport Protocol, INWG General Note 96, 1
[8] Chretien, G.J., et al., The SITA Network, Summary Description, Computer Comm. Network Conference, Univ. of Susex, Brighton, U.K., Sept. 1973
[9] Day, J.D., Resource Sharing Protocols, Computer, Sept. 1979
[10] DDCMP, Digital Data Communication Message Potocol, DEC, Maynard, Mass. 01754, Dec. 1974
[11] Deparis, M., The Implementation of and End-to-End Protocol by EIN Centers: A Survey and Comparision, Reseau Cyclades SCH 033, Aug. 1976
[12] Despres, R., et al., RCP: The Experimental Packet Switching Network of the PTT, Proc. of the AFCET Congress, Rennes, Nov. 1973
[13] Donnan, R.A., Kersey, J.R., SDLC: A Perspective, IBM Systems Journal vol. 15, no. 1, 1976
[14] EIN, An End-to-End Transfer Protocol for EIN, EIN/76/002, NPL, Middlesex, England, 1976
[15] Eisenbis, J.L., Conventions for Digital Data Communications Link Desing, IBM Systems Journal, vol. 6, no. 4, 1967
[16] Folt, H.C., Status Report on New Standards for DTE/DCE Interface Protocols, Computer, Sept. 1979
[17] Fuller, S.H., et. al., Multimicroprocessors: An Overview and Working Example, Proc. of the IEEE, vol. 66, no. 2, Feb. 1978
[18] García, M.H., Selga, J.M., Line Control Unit for a Packet Switching Network, MIMI78, Zurich 1978
[19] Gaudissart, E., et al., Description of Protocols Implementation in the SGB Network, Proc. of the Computer Network Protocols Symposium, Liege, 1978
[20] Gitman, I., Comparision of Hop-by-Hop and End-to-End Acknowledgment Schemes in Computer Communication Networks, IEEE Trans. on Communications, Nov. 1976
[21] Gray, J.P, Line Control Procedures, Proc. IEEE, vol. 60, no. 11, Nov. 1972
[22] IFIP WG.6.1, Proposal for a Standard Virtual Terminal Protocol, INWG protocol 91, 1978
[23] ISO, HDLC Frame Structures, ISO DIS 3309, 2, 1975
[24] ISO, High Level/Data Link Control Procedures, ISO/TC/SGG, N-1005, 1975
[25] Jensen, D., The Honeywell Experimental Distributed Processor. An Overview, Computer, Jan., 1978
[26] Mazare, G., MCS-A Symetric Multimicroprocessor System, Euromicro, Venecia, 1976
[27] Mazare, G., A Few Examples of How to Use a Symetric Multimicroprocessor, The 4th Annual Symposium on Computer Architecture, March 1977
[28] Naffah, N., High Level Protocol for Alphanumeric Data-Entry Terminals, Reseaux Cyclades, TER2538, 1., Nov.1977
[29] Pouzin, L., Presentation and Major Design Aspects of the Cyclades Computer Network, Proc. 3td Data Communications Symposium, Tampa Fla. 1973
[30] Roberts, L.G., Computer Network Development to Achieve Resource Sharing, SJCC 1972, AFIPS Press, Montvale NY 07645
[31] Scantlebury, R.A., Wilkinson, P.T., The National Physical Laboratory Data Communication Network, Proc. 2nd Intern. Conference on Computer Comm., ICCC Stockholm, 1974
[32] Scantlebury, R., End-to-End Transport Protocols, IRIA, March 1976
[33] Schicker, P., Zimmermann, H., Scroll Mode Virtual Terminal in EIN, 2nd European Workshop on Network Protocols, NPL, Oct. 1976
[34] Stenning, N.V. A data Transfer Protocol, Computer Networks 1, 1976
[35] Sunshine, C.A., Interprocess Communication Protocols for Computer Networks, TR 105 Digital Systems Lab. Standford Univ. 1975
[36] Swan, R.J., et al., CM*-A Modular Multimicroprocessor, AFIPS Conf. Proc., vol. 46, 1977
[37] Swan, R.J., et al., The Implementation of the CM* Multimicroprocessor, AFIPS Conf. Proc., vol. 46, 1977
[38] Tymes, L., TYMNET-A Terminal Oriented Communications Network, AFIPS Conf. Proc., vol. 38, 1971
[39] Valero, M.C. y otros, Sistemas Multimicroprocesadores: Características y Posibilidades, Novática, Nov./Dic. 1979
[40] Walden, D.C., The Evolution of Host-to-Host Protocol Technology, Computer, Sept. 1979
[41] Wecker, S., Computer Network Architecture, Computer, Sept.1979
[42] Wulf, W.A., Bell, G.C., Cmmp-A Multiminiprocessor, AFIPS Conf. Proc., vol. 41, 1972
[43] Zimmermann, H., The Cyclades End-to-End Protocol, Proc. 4th Data Comm. Symposium, ACM/IEEE, 1975

buses normalizados y normas de comunicación 3

E. Herrada, M. Medina, M. Valero

3.0 INTRODUCCIÓN

Este capítulo está dedicado a las normas o intentos de normalización que se han desarrollado en torno a la comunicación entre las partes de un microcomputador. Se han recopilado normas relativas a los tres aspectos que presenta esta comunicación.

Para la comunicación a nivel más interno, se han descrito tres buses:

MUBUS: Un intento de normalización a nivel europeo.

S100: Bus que ha pasado de los microcomputadores personales a norma internacional, propuesta por el IEEE.

MULTIBUS: Bus propuesto por un fabricante de microprocesadores, que gracias a la gran cantidad de segundas fuentes, se ha convertido casi en una norma, y que además permite trabajar a varios procesadores sobre el mismo bus.

Para la comunicación entre el procesador y los periféricos hemos considerado dos niveles de comunicación, según que la interfaz se encuentre próxima o lejana al procesador. Si está lejana, la comunicación debe admitir datos y su correspondiente sincronización, y además órdenes de control. Las normas más empleadas son las propuestas por el IEEE:

CAMAC: Pensada cuando los microprocesadores aún no habían llegado a los sistemas más sencillos.

IEEE 488/HPIB: Propuesta originalmente por un fabricante, ha llegado a convertirse en norma internacional gracias a su oportunidad.

Si la interfaz se encuentra próxima al procesador, la comunicación entre aquél y el periférico es más sencilla. Los procedimientos de comunicación más usados entre los microprocesadores son:

Bucle de 20 mA: Norma heredada de los terminales de telex, Teletype, cuando la electromecánica aún dominaba a la electrónica.

RS232: Norma EIA compatible con la V.24 de la CCITT, destinada a ser la única en vigor, por su mayor flexibilidad.

3.1 BUS NORMALIZADO EUROMICRO: MUBUS

Esta norma ha sido propuesta por J.D. Nicoud y J.P. Veuille en la revista de la asociación europea de usuarios de microprocesadores, EUROMICRO NEWSLETTER (vol. 1 n.º 3, 1975; vol.2 n.º2, 1976; vol. 3 n.º 1, 1977), y también en la editada por el laboratorio de calculadoras digitales de la E.P.F. Lausanne (Suiza), MICROSCOPE (vol. 1 núms. 2, 8). La norma ha sido adoptada por fabricantes suizos, italianos y españoles, y por varias universidades politécnicas, como las de Turín y Barcelona (E.T.S.I. Telecomunicación).

3.1.1 Descripción lógica de las señales

Las líneas del bus se clasifican en tres grupos:

—16 líneas de dirección y 16 de datos, activas altas (afirmadas).

—20 líneas de control, activas bajas (negadas).

—10 líneas de alimentación y masa.

3.1.1.1 *Líneas de datos y direcciones*

Estas líneas son tres-estados, activas altas (afirmadas) en el bus y normalmente amplificadas (buffered).

DIRECCIÓN (ADDRESS): 16 líneas emitidas por el amo (procesador o unidad de acceso directo a memoria, ADM [DMA]). Su valor se valida en el bus mediante una de las señales de control ADMEM, ADPER o REFRESH.

DATO (DATA): 16 líneas emitidas por el amo o el esclavo seleccionado. Su temporización se controla mediante las señales ADMEM, ADPER, WRITE, NODA e INTACK. Los tiempos de acceso de amo y esclavo (p. ej., memoria) deben ser acordes.

3.1.1.2 *Líneas de control*

Todas las señales de control, excepto las dos encadenadas, son activas bajas (lógica negativa o negadas) en el bus, se controlan mediante puertas colector abierto o tres-estados, salvo algunas.

Las señales de control son descritas funcionalmente en su forma afirmada y así se representan en los cronogramas (diagramas de tiempos). La inversión física de la señal sobre el bus se indica con el sufijo -LOW.

ADMEM (LOW): Dirección de memoria válida.

ADPER (LOW): Dirección de periférico válida.

REFRESH (LOW): Dirección de refresco de memoria dinámica válida.

NOTREADY (LOW): Petición de suspensión del ciclo del procesador. La señal recibe también los nombres de NOTRDY, NOTYET o STOP. Esta señal debe ser válida cuando es muestreada por el procesador. Su duración puede estar limitada (4 μs para el 6800). La relación entre la duración de NOTREADY y el incremento en el ciclo de acceso a memoria del procesador depende de éste.

WRITE (LOW): Define si se realiza un ciclo de lectura (WRITE = 0) o escritura (WRITE = 1).

NODA (LOW): El dato en el bus no es válido.

RESET (LOW): Inicializa el procesador y los periféricos. Debe ser un pulso de duración limitada en los sistemas que incluyen memorias dinámicas refrescadas por el procesador (Z80) (50 μs a 200 μs).

INTREQ (LOW): Línea de petición de interrupción. El dispositivo interruptor puede ser detectado por muestreo programado o mediante las señales INTACK y(o) INTIN e INTOUT.

INTACK (LOW): Lee el vector de interrupción del dispositivo con mayor prioridad, que haya solicitado la interrupción. Mientras INTACK es activa no se admiten interrupciones.

INTIN: Entrada de la cadena de prioridad de interrupción.

INTOUT: Salida de la cadena de prioridad de interrupción.

NMI (LOW): Petición de interrupción no enmascarable, normalmente reservada para caídas de tensión.

HOLDREQ (LOW): Petición del bus por un viceamo (procesador secundario o dispositivo ADM).

HOLDACK (LOW): Cesión del bus por el amo, el cual sólo sigue controlando esta línea.

HOLDIN: Entrada de la cadena de prioridad de petición del bus.

HOLDOUT: Salida de la cadena de prioridad de petición del bus.

PROCREQ (LOW): Petición de procesador, para uso en sistemas multiprocesador.

USERSCLOK: Reloj para temporizaciones y sincronización (1 Hz o 50 Hz).

SYSTEMCLOCK: Reloj del procesador, para uso en sistemas multiprocesador especiales (6800 invertidos de fase) o para manipulación de periféricos especiales.

3.1.1.3 *Líneas de alimentación*

En la mayoría de los sistemas de microprocesadores se usan 4 tensiones. El uso de convertidores CC-CC (DC-DC) permite reducir el número de líneas de alimentación en el bus; pero encarece las placas.

Además de la masa (GND) y la alimentación principal (5 V para procesador, memoria y periféricos), se requiere otra masa y ±15 V para el mínimo ruido en los convertidores A/D. Normalmente las dos masas sólo se cortocircuitan en la placa del convertidor A/D.

Si se usan acopladores ópticos, se debe usar una masa y alimentación separadas para la parte externa aislada, tal como muestra la figura 3.1.

En consecuencia las líneas de alimentación sugeridas son:

GND (0V)	:	Masa lógica para señales MUBUS y alimentaciones.
+ 5 V	:	Alimentación principal.
+ 12 V	:	Segunda alimentación principal, para algunos circuitos MOS y convertidores CC-CC.
− 5 V, − 12 V	:	Alimentaciones adicionales usadas por algunos circuitos MOS.
GND (A)	:	Masa separada para convertidores A/D.
+ 15 V, − 15 V	:	Alimentaciones para convertidores A/D solamente.
GND (X)	:	Masa separada para circuitos optoaislados externos.
+ 5 V (X)	:	Alimentación para circuitos optoaislados externos.

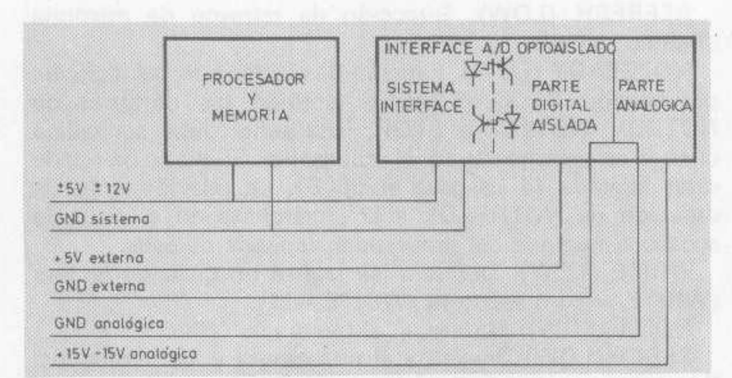


Figura 3.1 Distribución de las masas y alimentaciones del bus MUBUS en una unidad convertidora A/D optoaislada. En este esquema se aprecia la utilidad de cada una de las tres masas del sistema: lógica, externa y analógica.

3.1.2 Ciclos de acceso a memoria o periférico

La figura 3.2 muestra el estado que deben tomar las señales ADMEM, ADPER, WRITE, DATA, NODA y NOTREADY en los cuatro ciclos básicos de acceso a memoria o periférico.

La figura 3.3 muestra el ciclo de petición de interrupción, así como unos sencillos circuitos que permiten controlar las prioridades.

La figura 3.4 muestra el cronograma de la secuencia de petición y devolución del control del bus por en vice-amo (ADM) y por otro amo (multiprocesadores).

3.1.3 Especificaciones mecánicas

La figura 3.5 muestra la distribución recomendada de las líneas, así como las dimensiones de las placas, que coinciden con la norma Europa.

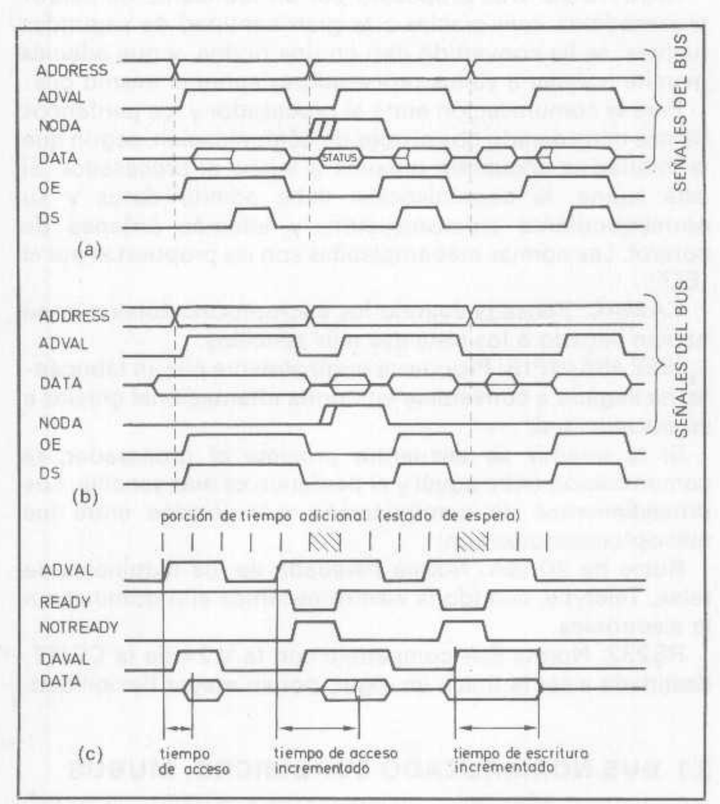


Figura 3.2 Cronograma de los ciclos de acceso a memoria y periférico en el bus MUBUS. Junto a las señales del bus se han incluido las que generan internamente los dispositivos emisor y receptor, para controlar los amplificadores tres-estados (OE) y para validar o cargar el dato (DS), respectivamente. La señal ADVAL hace referencia a las ADPER o ADMEM, según corresponda: a) Ciclo de lectura. La señal NODA impide que se coloque el dato leído en el bus antes de que el procesador lo espere. b) Ciclo de escritura. La señal NODA indica que el dato a escribir no está listo en el bus. c) Uso de la señal READY= NOTYET.LOW para añadir ciclos de espera en una operación de lectura o escritura. En la parte superior de la figura se han marcado los ciclos o fases del reloj del procesador.

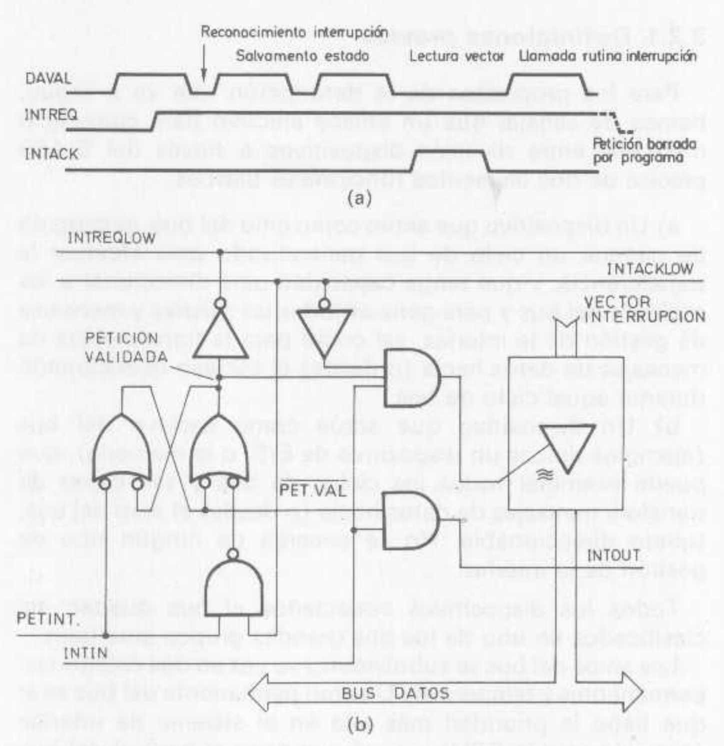


Figura 3.3 Petición y servicio de una interrupción. a) Cronograma. Cada vez que se activa DAVAL se hace una transferencia de datos por el bus. El número de ciclos para salvar el estado del procesador, así como el tiempo que se tarda en reconocer la interrupción (INTACK) dependen del procesador empleado. b) Sencillo esquema que permite establecer un sistema de prioridades en las interrupciones mediante un encadenado de las señales INTIN e INTOUT. Un sencillo biestable RS impide que la petición de interrupción interna (PETINT), llegue al bus mientras se está sirviendo otra (INTACK=1). La señal INTIN sólo "pasa" a INTOUT si no hay petición validada (PETICIÓN VALID=0); si la hay (PET.VAL=1), se corta la cadena (INTOUT=0) y se pone en el bus de datos el vector de interrupción.

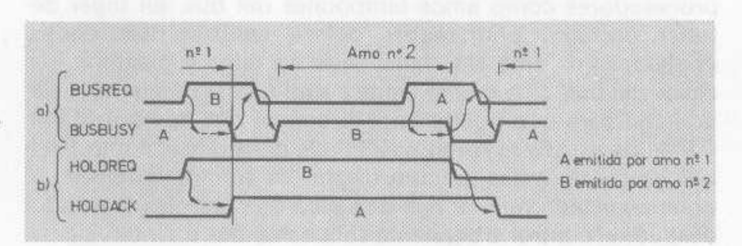


Figura 3.4. Cronograma de las señales necesarias para: a) Transferir el control del bus de un amo a otro (sistemas multiprocesadores). b) Transferir el control a un vice-amo (periférico ADM). En la figura *a* se ha sustituido la palabra HOLD por BUS, para adaptar el nombre de la señal a su significado real. Las señales empleadas en el bus para esta transferencia (*a*) son PROCREQ y HOLDACK. En ambos casos las prioridades se controlan mediante la cadena HOLDIN/HOLDOUT.

3.2 BUS NORMALIZADO S-100

Se trata de una norma de bus paralelo, útil para la comunicación entre módulos en alta velocidad. Es de aplicación a elementos de interfaz para componentes de sistemas computadores interconectados a través de un conjunto de 100 líneas en paralelo, que comprenden líneas para la transferencia simultánea de informaciones de tres tipos: direcciones, datos y controles. Cada subconjunto particular de hilos físicos (bus particularizado) transportará información correspondiente al ciclo específico de operación que esté en curso de ejecución.

Puede afirmarse que el bus S-100 es actualmente el más usado, siendo el empleado por más de un centenar de

Figura 3.5 Características físicas del sistema MUBUS. a) Tarjeta normalizada europea formato simple. b) Tarjeta normalizada europea formato doble. c) Distribución de señales recomendada por EUROMICRO. Obsérvese que es un subconjunto de las descritas en el texto. d) Distribución de señales recomendada por MICROSCOPE. La fila 7 del conector puede tener un tope para impedir la colocación de la tarjeta en sentido inverso (posición código).

fabricantes para más de 700 tipos diferentes de tarjetas y sistemas completos, aunque conviene advertir que no son completamente compatibles todos los que afirman serlo.

Su aplicación para sistemas microcomputadores (o para sus subsistemas) se refiere en particular a los que constan de un número total de dispositivos interconectados que no supere la veintena, usan una vía de transmisión que es eléctricamente corta (no influyen los retardos de propagación) y trabajan con velocidades máximas de transferencia bajas, con señales en el bus de hasta 6 MHz.

Esta norma trata de definir un sistema de interfaz de manejo fácil y racional, que asegure la compatibilidad de los diseños actuales y futuros sobre el bus S-100, con facilidad de ampliaciones modulares y de interconexión de dispositivos de fabricantes distintos para construir sistemas completos, con la menor limitación posible en el rendimiento de cada una de las partes componentes.

Históricamente este bus fue usado en primer lugar en los microcomputadores Altair por la firma MITS Inc., con diseños basados sobre el 8080. Actualmente se utiliza el bus S-100 para sistemas con otros microprocesadores (Z-80, 6800, 6502...) y es el de más amplio soporte comercial en el área de los microcomputadores.

El diseño de los primitivos sistemas que usaban S-100 fue anterior a la aparición de las pastillas controladoras para el 8080 (como la 8228), razón por la cual la norma original de este bus contiene gran cantidad de señales de control (39) que eran necesarias para cubrir la falta de patillas del circuito de Intel, pero que ya no resultan precisas en su totalidad.

A pesar de que la utilización del bus S-100 se ha hecho extensiva desde hace más de diez años, es de resaltar el hecho de que su estandarización estricta sólo ha sido propuesta recientemente, en base a los trabajos que una subcomisión del IEEE ha estado acometiendo desde 1976. La especificación preliminar resultante se presentó en junio de 1978, y posteriormente una propuesta modificada de esta norma se ha concluido de redactar durante 1979, con vistas a su presentación ante el IEEE Standards Board, encontrándose actualmente (1980) en fase de exposición y discusión pública por los interesados. Esta versión final, que se conoce como IEEE Task 696,1/D2, es la que se va a exponer en lo que sigue.

La nueva norma para el bus S-100 trata de eliminar algunos de los problemas que presentaba la primitiva, y de actualizar su nivel para que resulte apropiado para los microprocesadores de 16 bits. A tales fines se ha ampliado de 16 a 24 el número de líneas de direcciones, y el bus de datos que estaba formado por dos buses unidireccionales de 8 bits (uno para entrada y otro para salida de datos) se presenta ahora con la posibilidad adicional de constituir también un bus bidireccional de 16 líneas.

Se incluyen dos señales nuevas de diálogo (handshaking) para permitir la coexistencia de tarjetas de memoria sobre 8 y 16 bits, se han añadido 3 líneas más para alimentación y otra para fallos en la misma. Se han eliminado asimismo algunas señales que originariamente se reservaban para usos de panel frontal, pensando que tales líneas pueden manejarse mejor poniendo cables puenteadores desde la tarjeta de la CPU.

Especifica también la norma que aquí presentamos un protocolo para acceso directo a memoria (DMA), con posibilidad de que hasta 16 dispositivos puedan engancharse al bus con facultad de tener control del mismo, siempre que se incluya en el sistema el circuito de arbitraje correspondiente.

3.2.1 Definiciones previas

Para los propósitos de la descripción que va a seguir, hemos de señalar que un enlace efectivo para comunicar mensajes entre diversos dispositivos a través del S-100 precisa de dos elementos funcionales básicos:

a) Un dispositivo que actúe como amo del bus, encargado de generar un ciclo de bus generalizado para efectuar la transferencia, y que tenga capacidad para direccionar a los esclavos del bus y para generar todas las señales y mensajes de gestión de la interfaz, así como para la transferencia de mensajes de datos hacia (o desde) el esclavo direccionado durante aquel ciclo de bus.

b) Un dispositivo que actúe como esclavo del bus (ejemplos típicos un dispositivo de E/S, o la memoria), que puede examinar todos los ciclos de bus y sea capaz de transferir mensajes de datos hacia (o desde) el amo del bus, siendo direccionable. No se encarga de ningún tipo de gestión de la interfaz.

Todos los dispositivos conectados al bus quedan así clasificados en uno de los dos grandes grupos anteriores.

Los amos del bus se subdividen a su vez en dos categorías: permanentes y temporales. Un amo permanente del bus es el que tiene la prioridad más alta en el sistema de interfaz (normalmente la CPU) y es el que tiene el control del bus generalmente. Un amo temporal del bus es el que tiene que solicitar, mediante una operación de bloqueo (hold), al amo permanente que le ceda el control del bus durante un número variable de ciclos, para después devolvérselo. Denominamos un ciclo de DMA a la transferencia de control del bus desde la CPU a un amo temporal y el retorno correspondiente. No se considera la posibilidad de tener varios niveles de DMA.

En los casos en que coexistan en un mismo sistema varios procesadores como amos temporales del bus, en lugar de tener un amo permanente puede resultar más eficaz implantar un "amo simulado" que no dirija y gestione los ciclos del bus, sino que se limite a suministrar un intervalo de arbitraje para que pueda estabilizarse el bus de control del DMA. De esta forma el amo simulado consigue el control del bus entre las tomas de control por los amos temporales, y pone en estado nulo al bus de salida de controles, pasando después el control al siguiente solicitante tras un intervalo de arbitraje que dure un ciclo de reloj.

3.2.2 Señales del bus

Se va a presentar aquí una especificación que define las señales de este bus según la norma IEEE 696.1/D2 que es una extensión del bus original de Altair, cuya descripción escueta puede encontrarse en forma de tabla en el Apéndice III.1, y que pensamos que ha quedado superada. Señalemos, no obstante, que la mayoría de los sistemas existentes que se adaptaban a la norma primitiva son fácilmente compatibles con esta nueva norma que contiene propuestas para extensión de direcciones y datos. Con objeto de poder relacionar ambos estándares, se van a indicar en la tabla del Apéndice con un asterisco aquellas señales de nueva definición.

Las 100 posibles señales de este bus se normalizan numeradas según las patillas del 1 al 100, pero hay que indicar que sólo se ha especificado la función para 93 señales, quedando por lo tanto 7 sin usar en este momento (norma propuesta en 1979). Todas las señales, excepto las de alimentación, están limitadas a niveles positivos entre 0 y

+5 V y no pueden tener tiempos de subida o bajada (en carga) inferior a 5 ns.

Si distribuimos funcionalmente todo el conjunto de las señales del S-100 obtendremos la siguiente clasificación preliminar:

a) bus de datos, con 16 líneas (n.os 35, 36, 38 a 43 y 88 a 95)

b) bus de direcciones, con 16 líneas (n.os 29 a 34, 37 y 79 a 87) más otras 8 líneas (total 24) de ampliación (n.os 15 a 17, 59 y 61 a 64)

c) bus de control, con un total de 27 líneas:

c1) estado, con 8 líneas (n.os 44 a 48, 58, 96 y 97)

c2) salida de controles, con 5 líneas (n.os 25, 26 y 76 a 78)

c3) entrada de controles, con 6 líneas (n.os 3, 12, 60 y 72 a 74)

c4) control de D.M.A., con 8 líneas (n.os 14, 18, 19, 22, 23, 55 a 57)

d) bus de interrupciones vectorizadas, con 8 líneas (n.os 4 a 11)

e) bus de servicio, con un total de 18 líneas:

e1) alimentación con 9 líneas (n.os 1, 2, 20, 50 a 53, 70 y 100)

e2) reloj, con 2 líneas (n.os 24 y 49)

e3) reposición o "reset", con 3 líneas (n.os 54, 75, y 99)

e4) validación de escritura en memoria, con 1 línea (n.º 68)

e5) "esclavo fantasma", con 1 línea (n.º 67)

e6) condiciones especiales, con 2 líneas (n.os 13 y 98)

Quedan las líneas n.os 27, 28, 69 y 71 como reserva para uso posterior y las líneas n.os 21, 65 y 66 quedan sin definir y pueden ser utilizadas opcionalmente por cada fabricante individual con tal de que se suministren con cables puenteadores para evitar conflictos y que las señales en ellas se limiten a niveles lógicos de 5 V.

Se pasa ahora a describir, en los apartados siguientes, la naturaleza y utilización de cada uno de estos buses subconjuntos del S-100. Se incluye una tabla completa (tabla 3.1) que reúne ordenadamente todas las señales del bus S-100 y su función resumida más otros detalles de interés.

3.2.2.1 *Bus de datos*

Consta de 16 líneas que se agrupan en forma de 2 buses unidireccionales de 8 bits para poder operar con octetos, y en forma de bus único bidireccional para poder operar con palabras de 16 bits. (La primitiva norma no incluía esta segunda realización del bus de datos).

La entrada o salida de datos se especifica siempre con relación al amo del bus en ese momento; cuando es el amo del bus quien transmite hacia un esclavo del bus, se dice que hay datos de salida, mientras que en caso inverso se habla de datos de entrada.

En el funcionamiento con octetos las transferencias de información (datos) se hacen mediante dos buses unidireccionales, cada uno de 8 bits; la salida aparece en hilos DO7 a DO0, en este orden de mayor a menor en el peso de los bits; la entrada, por su parte, se hace sobre los hilos DI7 a DI0 con el mismo criterio de pesos.

En el funcionamiento con palabras de 16 bits se crea un bus único bidireccional con las 16 líneas DI y DO. Para ello se utilizan dos señales de control, sXTRQ para petición y SIXTN para aceptación (acknowledge). Se consigue tener

n.º del terminal	nombre de la señal (y tipo)		nivel activo/ colector abierto	*Función y características de la señal*
1	+8 V	(B)		Mínimo instantáneo (MINI) superior a 7 V. Máximo instantáneo (MAXI) inferior a 25 V. Máximo promedio (MAXP) inferior a 11 V.
2	+16 V	(B)		MINI > 14.5 V. MAXI < 35 V (ver n.º 1) MAXP < 21.5 V.
3	XRDY	(S)	H/no	Una entrada de «preparado» al amo actual del bus. Activada junto con la n.º 72 pasa el bus a «preparado»
4	VI0	(S)	L/si	Señal 0 del vector de interrupciones.
5	VI1	(S)	L/si	Señal 1 del vector de interrupciones
6	VI2	(S)	L/si	Señal 2 del vector de interrupciones.
7	VI3	(S)	L/si	Señal 3 del vector de interrupciones.
8	VI4	(S)	L/si	Señal 4 del vector de interrupciones.
9	VI5	(S)	L/si	Señal 5 del vector de interrupciones.
10	VI6	(S)	L/si	Señal 6 del vector de interrupciones.
11	VI7	(S)	L/si	Señal 7 del vector de interrupciones
12	NMI	(S)	L/si	Interrupción no enmascarable.
13	PWRFAIL	(B)	L/no	Señal de fallo de alimentación al bus.
14	DMA3	(M)	L/si	Bit 3 de prioridad para amo temporal del bus (el bit más significativo).
15	A18	(M)	H/no	Bit 18 de dirección ampliada.
16	A16	(M)	H/no	Bit 16 de dirección ampliada.
17	A17	(M)	H/no	Bit 17 de dirección ampliada.
18	SDSB	(M)	L/si	Control para inhabilitar a las 8 señales de estado.
19	CDSB	(M)	L/si	Control para inhabilitar a las 5 señales de salida de controles.
20	GND	(B)		Masa del sistema (común con la n.º 100)
21	NDEF			Sin especificar (no definida)
22	ADSB	(M)	L/si	Control para inhabilitar a las 16 señales de dirección.
23	DODSB	(M)	L/si	Control para inhabilitar a las 8 señales de salida de datos.
24	Φ	(B)	H/no	Señal de origen de temporización para el bus.
25	pSTVAL	(M)	L/no	Validación («strobe») de estado.
26	pHLDA	(M)	H/no	Control que junto con HOLD (n.º 74) coordina las operaciones de transferencia de los amos de bus.
27	RFU			Sin especificar (reservada para usos posteriores).
28	RFU			Sin especificar (reservada para usos posteriores)
29	A5	(M)	H/no	Bit 5 de dirección.

Tabla 3.1 Bus S-100 (IEEE Task 696,1/D2)

n.º del terminal	nombre de la señal (y tipo)		nivel activo/ colector abierto	*Función y características de la señal*
30	A4	(M)	H/no	Bit 4 de dirección.
31	A3	(M)	H/no	Bit 3 de dirección.
32	A15	(M)	H/no	Bit 15 de dirección.
33	A12	(M)	H/no	Bit 12 de dirección.
34	A9	(M)	H/no	Bit 9 de dirección.
35	DO1 DATA1	(M) (M-S)	H/no	Bit 1 de dato de salida/Bit 1 de dato bidireccional.
36	DO0 DATA0	(M) (M-S)	H/no	Bit 0 de dato de salida/Bit 0 de dato bidireccional
37	A10	(M)	H/no	Bit 10 de dirección-
38	DO4 DATA4	(M) (M-S)	H/no	Bit 4 de dato de salida/Bit 4 de dato bidireccional.
39	DO5 DATA5	(M) (M-S)	H/no	Bit 5 de dato de salida/Bit 5 de dato bidireccional.
40	DO6 DATA6	(M) (M-S)	H/no	Bit 6 de dato de salida/Bit 6 de dato bidireccional.
41	DI2 DATA10	(S) (M-S)	H/no	Bit 2 de dato de entrada/Bit 10 de dato bidireccional.
42	DI3 DATA11	(S) (M-S)	H/no	Bit 3 de dato de entrada/Bit 11 de dato bidireccional.
43	DI7 DATA15	(S) (M-S)	H/no	Bit 7 de dato de entrada/Bit 15 de dato bidireccional.
44	sM1	(M)	H/no	Indica que el ciclo en curso es de extracción del código de operación.
45	sOUT	(M)	H/no	Identifica el ciclo de transferencia de datos hacia un dispositivo de salida.
46	sINP	(M)	H/no	Identifica el ciclo de transferencia de datos desde un dispositivo de entrada.
47	sMEMR	(M)	H/no	Identifica los ciclos de bus en que se transfieren datos desde memoria hacia un amo de bus y que no sean ciclos de extracción de instrucción de reconocimiento de interrupción.
48	sHLTA	(M)	H/no	Señal de reconocimiento de la ejecución de HLT.
49	CLOCK	(B)		De 2 MHz (±10 kHz) y ciclo de trabajo del 40 al 60 %. No es preciso que se sincronice con ninguna otra señal del bus.
50	GND	(B)		Masa del sistema (común con la n.º 100)
51	+8 V	(B)		(ver n.º 1)
52	−16 V	(B)		MAXI<.14,5 V.; MINI>.35 V; MINP>21,5 V. (ver n.º 1).
53	GND	(B)		Masa del sistema (común con la n.º 100).
54	SLAVE CLR	(B)	L/si	Señal de reinicialización para los esclavos del bus. Puede activarse externamente.
55	DMA0	(M)	L/si	Bit 0 de prioridad para amo temporal del bus.
56	DMA1	(M)	L/si	Bit 1 de prioridad para amo temporal del bus.

Tabla 3.1 (Continuación)

n.º del terminal	nombre de la señal (y tipo)		nivel activo/ colector abierto	*Función y características de la señal*
57	DMA2	(M)	L/si	Bit 2 de prioridad para amo temporal bus.
58	sXRQ	(M)	L/no	Solicitud de transferencia sobre 16 bits (ver n.º 60).
59	A19	(M)	H/no	Bit 19 de dirección ampliada.
60	SIXTN	(S)	L/si	Respuesta afirmativa de los esclavos del bus a la señal de petición sXTRQ (n.º 58).
61	A20	(M)	H/no	Bit 20 de dirección ampliada.
62	A21	(M)	H/no	Bit 21 de dirección ampliada.
63	A22	(M)	H/no	Bit 22 de dirección ampliada.
64	A23	(M)	H/no	Bit 23 de dirección ampliada.
65	NDEF			Sin especificar (no definida).
66	NDEF			Sin especificar (no definida).
67	PHANTOM	(M-S)	L/si	Habilitar a los dispositivos «esclavos fantasma» (útil en principio para sistemas «bootstrapping» sin circuitos de panel frontal)
68	MWRT	(B)	H/no	Escritura en memoria; debe seguir a la pWR con un máximo de 30 ns de retraso (ver n.º 77).
69	RFU			Sin especificar (reservada para usos posteriores).
70	GND	(B)		Masa del sistema (común con la n.º 100).
71	RFU			Sin especificar (reservada para usos posteriores).
72	RDY	(S)	H/si	Ver señal n.º 3.
73	INT	(S)	L/si	Señal principal de petición de interrupción.
74	HOLD	(M)	L/si	Control para coordinar operaciones de transferencia de amo del bus. (ver n.º 26).
75	RESET	(B)	L/si	Reinicialización de los amos del bus. Puede generarse externamente. Precisa la activación de la POC (n.º 99)
76	pSYNC	(M)	H/no	Control que identifica el estado inicial, BS_1, de un ciclo de bus.
77	pWR	(M)	L/no	Control que indica la presencia de datos válidos en el bus de datos de salida.
78	pDBIN	(M)	H/no	Control que solicita datos al bus de entrada de datos.
79	A0	(M)	H/no	Bit 0 de dirección (el menos significativo).
80	A1	(M)	H/no	Bit 1 de dirección.
81	A2	(M)	H/no	Bit 2 de dirección.
82	A6	(M)	H/no	Bit 6 de dirección.
83	A7	(M)	H/no	Bit 7 de dirección.
84	A8	(M)	H/no	Bit 8 de dirección.
85	A13	(M)	H/no	Bit 13 de dirección.
86	A14	(M)	H/no	Bit 14 de dirección.
87	A11	(M)	H/no	Bit 11 de dirección.

Tabla 3.1 (Continuación)

n.º del terminal	nombre de la señal (y tipo)	nivel activo/ colector abierto	*Función y características de la señal*
88	DO2 (M) DATA2 (M-S)	H/no	Bit 2 de datos de salida/Bit 2 de dato bidireccional
89	DO3 (M) DATA3 (M-S)	H/no	Bit 3 de dato de salida/Bit 3 de dato bidireccional
90	DO7 (M) DATA7 (M-S)	H/no	Bit 7 de dato de salida/Bit 7 de dato bidireccional
91	DI4 (S) DATA12 (M-S)	H/no	Bit 4 de dato de entrada/Bit 12 de dato bidireccional.
92	DI5 (S) DATA13 (M-S)	H/no	Bit 5 de datos de entrada/Bit 13 de dato bidireccional.
93	DI6 (S) DATA14 (M-S)	H/no	Bit 6 de dato de entrada/Bit 14 de dato bidireccional.
94	DI1 (S) DATA9 (M-S)	H/no	Bit 1 de dato de entrada/Bit 9 de dato bidireccional.
95	DI0 (S) DATA8 (M-S)	H/no	Bit 0 (el menos significativo) de dato de entrada/Bit 8 de dato bidireccional.
96	sINTA (M)	H/no	Identifica el ciclo (s) de entrada que puede(n) seguir a la aceptación de una petición de interrupción.
97	sWO (M)	L/no	Identifica un ciclo de bus en el que se transfieren datos desde un amo de bus hacia un esclavo.
98	ERROR (S)	L/si	Significa que hay alguna condición de error durante el ciclo de bus en curso.
99	POC (B)	L/no	Señal de puesta a cero, al conectar alimentación para todos los dispositivos; cuando pasa al nivel bajo debe permanecer así al menos 10 ms.
100	GND (B)		Masa del sistema.

NOTAS:

H: significa señal en estado alto
L: significa señal en estado bajo
M: significa señal que debe generar el amo del bus cuando tiene el control del bus
S: significa señal que debe generar el esclavo del bus (las necesarias para la comunicación en curso)
B: las demás señales (algunas de ellas puede generarlas un amo)

Por tratarse de especificaciones no definitivas, estas señales podrían sufrir alguna pequeña modificación en el futuro.

Tabla 3.1 (Continuación)

un bus desde DATA 15 a DATA 0 formado por los hilos DI7, ..., DI0, DO7, ..., DO0 cuando ambas señales de control están en nivel bajo (su estado lógico VERDADERO). Veremos más adelante la especificación detallada del protocolo para 8/16 bits.

3.2.2.2 *Bus de direcciones*

Consta de 16 o 24 líneas en paralelo que sirven para que las señales que por ellas viajen, seleccionen una posición concreta de la memoria, o también un dispositivo concreto de E/S. Se puede proceder entonces a efectuar una comunicación durante el correspondiente ciclo de bus.

Al menos 16 bits de dirección son los que deben mantener (assert) todos los amos del bus (líneas A15-A0); pero pueden alcanzarse 24 bits de dirección si se desea, utilizándose para ello la facilidad de ampliación con otras 8 líneas (A23-A16). (Esta última posibilidad no existía con la norma primitiva).

De acuerdo con lo anterior, tenemos que el bus estándar de direccionamiento en memoria consta de 16 líneas que permiten especificar 64 K posiciones diferentes en memoria. Por su parte tenemos el bus extendido que consta de 24 líneas, en cuyo caso se pueden direccionar 16 M posiciones diferentes en memoria (A23 es el bit más significativo y A0 el menos significativo).

En cuanto al direccionamiento de dispositivos de E/S se tiene el modo estándar que emplea el bus A7-A0 permitiendo la especificación de hasta 256 dispositivos, y el modo extendido que emplea las líneas A15-A0 con lo que pueden direccionarse hasta 65536 dispositivos de E/S.

Con esta disposición de las líneas de direcciones se simplifica la ampliación en el número de dispositivos de E/S, que resultaría menos recomendable si se siguiese el direccionamiento tradicional que reservaba las líneas A15-A8 para asignar direcciones a tales dispositivos.

3.2.2.3 *Bus de control*

3.2.2.3.1 Bus de estado

Consta de 8 líneas que individualizan la naturaleza del ciclo de bus en curso y que califican el tipo de dato utilizado.

Todas las señales de estado se representan mediante nombres simbólicos que comienzan con una *s* minúscula. Las ocho señales de este bus, que deben ser generadas por el amo del bus en cada momento, son las siguientes:

Línea n.º		
44 -----	sM1	(Op-code fetch): Extracción del código de operación.
45 -----	sOUT	(Output): Salida.
46 -----	sINP	(Input): Entrada.
47 -----	sMEMR	(Memory Read): Lectura en memoria.
48 -----	sHLTA	(Halt Acknowledge): Reconocimiento de parada.
58 -----	sXTRQ	(16-bit Data Transfer Request): Petición de transferencia de datos de 16 bits.
96 -----	sINTA	(Interrupt Acknowledge): Reconocimiento de interrupción.
97 -----	sWO	(Write cycle): Ciclo de escritura.

Existe una señal de estado importante que no está directamente disponible sobre este bus, y que es la de escritura en memoria (Status Memory Write). Para crearla podemos combinar dos de las anteriores, y la definimos mediante la siguiente ecuación lógica:

Estado escritura en memoria = $\overline{\text{sOUT}} \cdot \text{sWO}$.

Se resume en la tabla 3.2 la definición de la palabra de estado (valor lógico de los 8 bits de estado) para cada uno de los ciclos de bus posibles.

3.2.2.3.2 Bus de salida de controles

Son 5 líneas que determinan el secuenciamiento y el movimiento de los datos durante cualquier ciclo del bus. Se utilizan para las señales de este bus nombres simbólicos que siempre comienzan por una *p*:

Línea n.º		
25 -----	pSTVAL	(Status Valid Strobe); Validación del estado, en combinación con

pSYNC indica que pueden muestrearse la dirección y el estado, estables ya sobre el bus en ese ciclo en curso.

26 ----- pHLDA (Hold Acknowledge); Reconocimiento de bloqueo, señala al amo temporal con la prioridad más alta en ese momento que el amo permanente está cediendo el control del bus.

76 ----- pSYNC (Sync.); Sincronismo, indica el comienzo de un nuevo ciclo de bus.

77 ----- pWR (Write); Escritura, es una señal de validación de escritura en general, para poder escribir datos desde el bus hacia un esclavo direccionado. El amo del bus puede activar esta señal tras completarse la duración de la señal pSYNC. Los datos pueden validarse con uno cualquiera de los flancos de pWR. La información sobre el estado y la dirección debe mantenerse durante cierto período de tiempo a partir del flanco de bajada de pWR.

78 ----- pDBIN (Data Bus In); Entrada al bus de datos, es una señal de validación de lectura generalizada, para poder pasar datos al bus desde un esclavo direccionado. El amo del bus la activa tras un tiempo mínimo especificado desde la activación de pSTVAL. Se desactiva, devolviendo así el bus de datos al estado de alta impedancia, poco antes de que puedan cambiar los buses de direcciones y estado

3.2.2.3.3 Bus de entrada de controles

Son 6 líneas que permiten a los esclavos del bus sincronizar, de acuerdo con sus propias condiciones internas, las operaciones de los dueños del bus. También les permiten la petición de operaciones a realizar por el amo del bus (p. ej., una interrupción).

Las señales son las siguientes:

Línea n.º
3 ----- XRDY (Señal especial de preparado)
12 ----- NMI (Interrupción no enmascarable)
60 ----- SIXTN (Aceptación de datos de 16 bits)
72 ----- RDY (Señal general de preparado)
73 ----- INT (Petición de interrupción)
74 ----- HOLD (Petición de bloqueo)

Vamos a describir a continuación el funcionamiento de todas ellas. Las señales XRDY y RDY son las de preparado (ready). Permiten la suspensión de ciclos de bus y la inserción de estados de espera (wait) hasta que ambas señales estén en valor alto, de forma que sirven para que los esclavos del bus puedan sincronizar según su propia velocidad de respuesta la operación de los amos del bus.

La señal XRDY es la usada generalmente por dispositivos

ciclos del bus \ señales de estado	*sM1*	*sOUT*	*sINP*	*sMEMR*	*sHLTA*	*sXTRQ*	*sINTA*	*sWO*
Lect. en Mem.	L	L	L	H	L	H(O) L(P)	L	H
Extrac. cód. op.	H	L	L	H	L	H(O) L(P)	L	H
Escrit. en Mem.	L	L	L	L	L	H(O) L(P)	L	L
Salida	L	H	L	L	L	H(O)	L	L
Entrada	L	L	H	L	L	H(O) L(P)	L	H
Recon. de Interr.	X	L	L	L	L	H(O) L(P)	H	H
Recon. de Parada	X	L	L	X	H	X	L	H

donde:
H representa el nivel Alto de la señal
L representa el nivel Bajo de la señal
X representa el valor indiferente
(O) se refiere a operaciones sobre 8 bits (octetos)
(P) se refiere a operaciones sobre 16 bits (palabras)
Donde no se hace la distinción entre (O) y (P) significa que coincide en ambos casos el valor del bit correspondiente.

Tabla 3.2

en el panel frontal para detener y hacer funcionar paso a paso a los amos del bus. Debe notarse que no ha de ser usada por más de un esclavo a la vez, ya que podría aparecer situación de conflicto por no ser una línea en colector abierto.

Las señales INT y NMI son las de interrupción, permiten la petición de servicio al amo permanente del bus.

La señal INT, que se implanta como una señal de comunicación entre el controlador de interrupciones y un amo del bus, si es aceptada por el amo del bus puede responderla con un ciclo de reconocimiento de interrupción y disponerse a adquirir información vectorizada del bus de datos. Es posible tener más de una petición de interrupción en el vector de interrupciones, realizando una «interrupción de grupo» con la señal INT combinada con el bus de interrupción vectorizada (ver apartado 3.2.2.4).

El amo del bus puede enmascarar la señal INT (generalmente se hace por software).

La señal NMI es una petición de interrupción que el amo del bus no puede enmascarar; al ser reconocida una señal NMI no es preciso generar un ciclo de aceptación de interrupción. La información significativa la lleva el flanco descendente de NMI, mientras que la INT actúa como una señal de nivel. La señal NMI es una entrada opcional de control para los amos del bus, y es prioritaria frente a otras posibles peticiones de interrupción.

La señal HOLD es usada por los amos temporales del bus para solicitar el control del bus al amo permanente. Puede enmascararse esta señal si el amo permanente del bus quiere impedir que otros dispositivos accedan a controlar el bus.

La señal SIXTN sirve de respuesta a la señal sXTRQ descrita en el bus de estado. Indica que es posible la transferencia de datos de 16 bits que se solicitó.

3.2.2.3.4 Bus de control de DMA

Se compone de 8 líneas que se utilizan en combinación con las señales HOLD y pHLDA anteriormente descritas. Permiten efectuar una transferencia ordenada del control del bus mediante el arbitraje entre peticiones simultáneas procedentes de los amos temporales del bus, así como la inhabilitación de los excitadores de señal del amo permanente.

Las 8 señales para control de acceso directo a memoria (DMA) son:

Cuatro señales para la inhabilitación de los excitadores de línea del amo permanente del bus durante una operación de DMA:

Línea n.º		
18 -----	SDSB	(Status Disable), Inhabilitar estado.
19 -----	CDSB	(Control Disable), Inhabilitar salida de controles.
22 -----	ADSB	(Address Disable), Inhabilitar direcciones.
23 -----	DODSB	(Data Out Disable), Inhabilitar salida de datos.

Otras cuatro señales DMA0, DMA1, DMA2 y DMA3 para efectuar la función de arbitraje del control del bus, ya que en cada momento codifican el número correspondiente al solicitante con más alta prioridad. La descripción del protocolo de arbitraje del DMA puede encontrarse en § 3.2.3.3.

3.2.2.4 *Bus de interrupciones vectorizadas*

Consta de 8 líneas y sus señales se utilizan en combinación con la señal INT para conseguir arbitrar entre ocho niveles de petición de interrupción. El funcionamiento típico es que estas señales sirvan de entrada a un esclavo del bus que enmascara y da prioridades a las peticiones, activa la solicitud de interrupción generalizada para el amo del bus, y cuando se entra en el ciclo de aceptación de la interrupción responde con los datos apropiados.

Las señales se denominan VI0-7 con prioridades descendentes desde la VI0 a la VI7 (líneas n.º 4 a 11).

Puesto que deben mantenerse activas hasta que se recibe la operación solicitada, se implantan como señales de nivel.

3.2.2.5 *Bus de servicio*

Es el conjunto de líneas para diversas utilidades.

3.2.2.5.1 Bus de alimentaciones del sistema

En la norma S-100 la alimentación se distribuye a los dispositivos en forma de tensiones continuas no reguladas. Se sigue, pues, la idea de regulación localizada en cada módulo, con lo que se simplifica la distribución pero, en cambio, se precisan dispositivos adicionales.

Hay un total de 9 líneas de alimentaciones:
a) dos para +8 voltios (líneas n.^{os} 1 y 51)
b) una para +16 voltios (línea n.º 2)
c) una para — 16 voltios (línea n.º 52)
d) cinco para masa (líneas n.^{os} 20, 50, 53, 70 y 100)

Debido a que están contiguas las líneas de 8 V con 16 V y con la de —16 V, una causa de posibles problemas son los cortocircuitos entre ellas al poner y quitar repetidas veces las tarjetas del sistema; por esta razón se recomienda usar resistores de descarga rápida para las tres líneas de tensión.

Puede apreciarse que las líneas de masa están distribuidas (según sus números que corresponden con la posición en el conector) de manera que las masas de baja impedancia se encuentran disponibles en ambos extremos y en las dos caras de la tarjeta.

3.2.2.5.2 Relojes

La señal ϕ (de la línea 24) es el reloj del sistema y es generada por el amo permanente del bus; nunca se transfiere durante una operación de intercambio por el bus. De esta señal de reloj es de donde deben tomarse todas las temporizaciones de control para los ciclos del bus, sea quien sea el dispositivo que en cada momento tenga el control.

La señal CLOCK (reloj, sobre la línea 49) viene especificada en esta norma como una señal de frecuencia 2 MHz, con 0,5 % de tolerancia, y sin relación alguna con cualquier otra señal del bus. Puede usarse para circuitos contadores, generadores de velocidad de transmisión, etc.

3.2.2.5.3 Funciones de reinicialización

Hay tres señales específicas para reinicializar el sistema:
----- RESET(línea n.º 75), que reinicializa a todos los amos del bus.
----- SLAVE CLR (n.º 54), para todos los esclavos del bus.
----- POC (n.º 99), «Power-on Clear»; Puesta a cero al conectar la alimentación al sistema que es cuando esta señal se activa. Es necesaria para hacer valer a las dos anteriores.

3.2.2.5.4 Validación de escritura en memoria

Mediante la señal MWRT (línea n.º 68) que debe generarse en un solo punto del sistema y ser válida para todos los amos del bus.

La ecuación lógica a cumplir es:

$$MWRT = pWR \cdot \overline{sOUT}$$

Es una señal que sólo se activa para los ciclos de escritura precisamente en memoria. La señal pWR, en cambio, se activa para cualquier ciclo de escritura.

3.2.2.5.5 «Esclavos Fantasmas»

Existe la señal PHANTOM (línea n.º 67) para asignar esclavos del bus a una posición común de la memoria. Al activarse se habilitan los esclavos fantasmas y se inhabilitan los esclavos normales.

3.2.2.5.6 Condiciones especiales

La línea 98 de S-100 es una línea general de error que se activa cuando ocurre un error de cualquier tipo (p. ej., de paridad, de escritura en memoria protegida, etc.) durante el ciclo de bus en curso. Toda la información de importancia relativa al origen del error debe guardarse durante el flanco descendente de esta señal.

La línea 13 soporta la señal PWRFAIL, que indica fallo inminente en la alimentación. Permanece en nivel bajo (activada) hasta que se restaura la tensión y la señal POC se vuelve a activar. Está especificado que debe activarse esta línea al menos 50 ms antes de que los reguladores locales de alimentación se salgan de sus valores límite establecidos.

3.2.3 Protocolos y secuenciamientos

Vamos a presentar aquí las principales características de

temporización, sincronización y arbitraje haciendo uso de las señales del S-100.

3.2.3.1 *Protocolo de transferencia de datos en 8 ó 16 bits*

Se trata de que pueden coexistir amos del bus para 8 bits y amos y esclavos para 16 bits en un mismo sistema. Para transferencias en 16 bits se unen los dos buses unidireccionales de 8 bits y se forma un único bus de datos bidireccional de 16 bits. Para efectuar el control de esta reunión existen 2 líneas asignadas: sXTRQ y SIXTN.

El amo actual del bus solicita una transferencia de datos de 8 bits no activando la señal sXTRQ. La salida de un octeto de datos desde el amo hacia el esclavo direccionado se lleva a cabo a través de las líneas del bus DO0-7. La entrada de un octeto de datos desde el esclavo direccionado hacia el amo actual del bus se lleva a cabo a través de las líneas DI0-7.

El amo actual del bus solicita una transferencia de datos de 16 bits activando la señal sXTRQ; si el esclavo direccionado está capacitado para transferencias sobre 16 bits activa la señal SIXTN y entonces se realiza la comunicación a través de las líneas del bus DATA0-15.

En cuanto a las transferencias sobre memoria para los dispositivos de memorización que puedan trabajar sobre 8 y 16 bits, la activación de unos bancos de memoria u otros puede ser separada o conjunta según el estado de la línea sXTRQ.

Cuando no se active la señal sXTRQ, las referencias a memoria indican transferencias de octeto con direccionado a través de las líneas A1-15 (o A1-23 si se amplía) mientras la línea A0 selecciona el octeto de mayor o menor peso. En cambio, cuando el amo del bus active la señal sXTRQ y el esclavo del bus la señal SIXTN, las referencias a memoria son para transferencias de octeto doble con direccionado a través de las líneas A1-15 (o A1-23) y sin intervención de la A0.

Cuando la línea SIXTN no se activa pasado un cierto período de tiempo después de la solicitud de transferencias

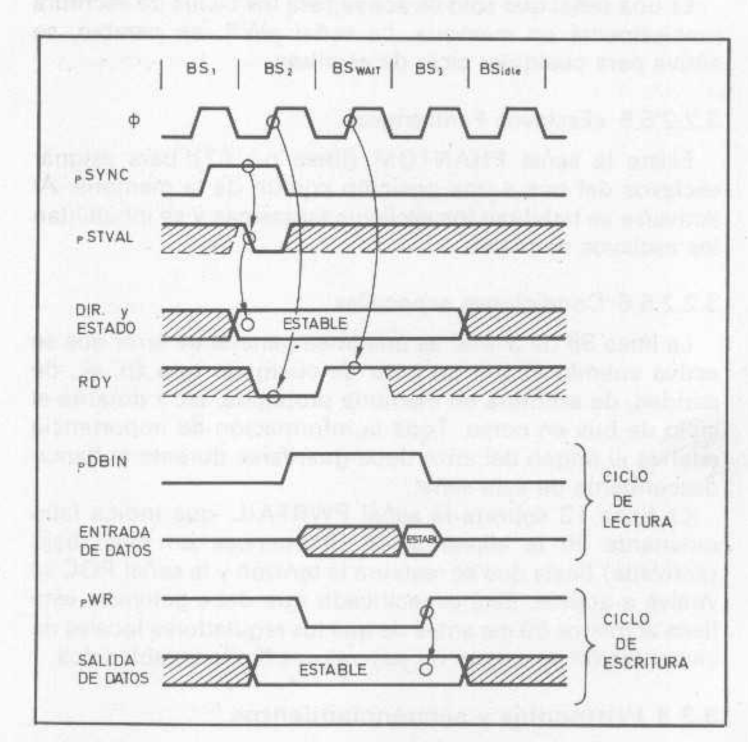


Figura 3.6 Cronogramas fundamentales del ciclo de bus. BS = Estados del bus.

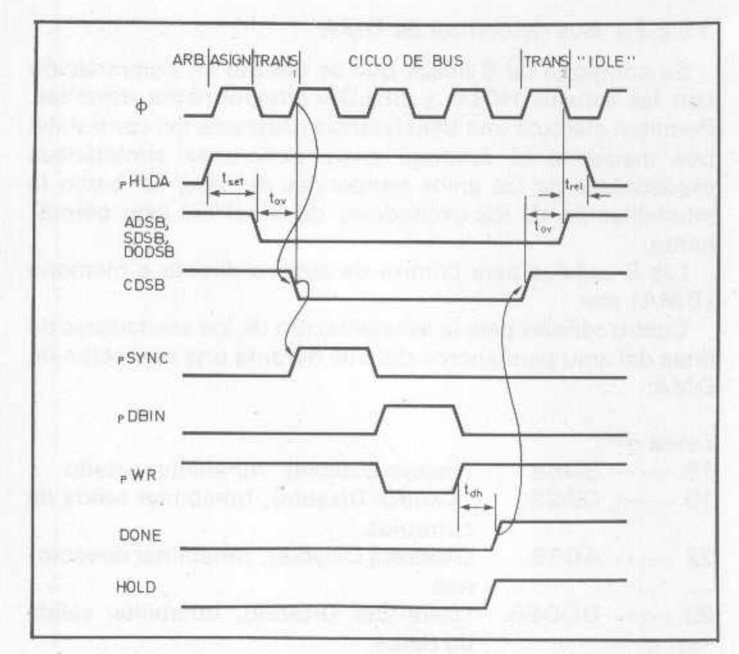


Figura 3.7 Cronogramas fundamentales de una transferencia de bus y un ciclo de DMA.

sobre 16 bits (por ejemplo: si el esclavo no puede aceptarla), los amos del bus pueden incluir circuitería para que la transferencia solicitada se efectúe mediante dos operaciones consecutivas con octetos mientras se deja el amo del bus en estado de espera. Si no existe tal circuitería resultará inmediatamente una condición de error, activándose la señal ERROR.

3.2.3.2 *Temporización del ciclo normal de bus*

Se trata de un ciclo pseudosincronizado, ya que el secuenciamiento de las señales de control guarda una relación específica con el reloj del sistema, ϕ.

En la figura 3.6 se muestra la temporización fundamental para un ciclo normal de bus, con un solo estado de espera intercalado por el esclavo direccionado.

3.2.3.3 *Protocolos para DMA*

Hay que permitir el solapamiento de las señales de control al comienzo y al final de la transición entre el control del bus por amos permanentes y temporales, para conseguir que el bus se estabilice antes de que se transfieran las informaciones y se reduzca así el ruido de impulsos espurios. Se considera en este sistema de interfaz que sólo puede haber un amo permanente del bus.

3.2.3.3.1 Protocolo de transferencia del control del bus

La operación de transferencia del control del bus para un solo ciclo de DMA debe realizarse de acuerdo con las relaciones fundamentales de secuenciamiento temporal que pueden verse en la figura 3.7, donde:

t_{set} = tiempo mínimo para que pueda completarse el ciclo de bus precedente.

t_{ov} = tiempo mínimo que tanto el amo temporal como el amo permanente deben excitar las señales de salida del control para asegurar una transferencia de control correcta.

t_{dh} = tiempo que debe transcurrir antes de que se

active la señal DONE (señal interna del amo temporal) para que se estabilicen las señales de salida de datos, estado y dirección en el ciclo normal del bus.

t_{rel} = tiempo mínimo que debe transcurrir para completarse la transferencia inversa del control del bus antes de que la señal pHLDA se desactive.

3.2.3.3.2 Protocolo de arbitraje entre solicitantes simultáneos del control del bus

El sistema de arbitraje utiliza 4 líneas del bus (DMA0-3) para decidir entre un máximo de 16 amos temporales, cada uno de los cuales tiene asignado un número único de prioridad en binario. El proceso continuo de arbitraje es asíncrono paralelo (no secuencial bit a bit).

Las relaciones de secuenciamiento temporal pueden verse en la figura 3.8, que muestra unos ejemplos concretos de temporización para un circuito específico de arbitraje. El primer cronograma corresponde al caso en que el solicitante no tiene competencia de ningún otro; y el segundo cronograma se refiere al caso en que un solicitante espera que el bus quede libre, mostrándose un arbitraje del bus con resultado de ganar el control y con el resultado opuesto.

Independientemente del diseño concreto del circuito de arbitraje, las reglas que debe cumplir un amo temporal del bus S-100 son las siguientes:

1) Si la señal HOLD no está activada, puede pasarla a activa si la señal pHLDA está en nivel bajo.

2) Debe desactivar HOLD cuando suba pHLDA, si es que otro controlador hace valer una prioridad más alta.

3) Debe desactivar HOLD siempre que deje de necesitar el bus.

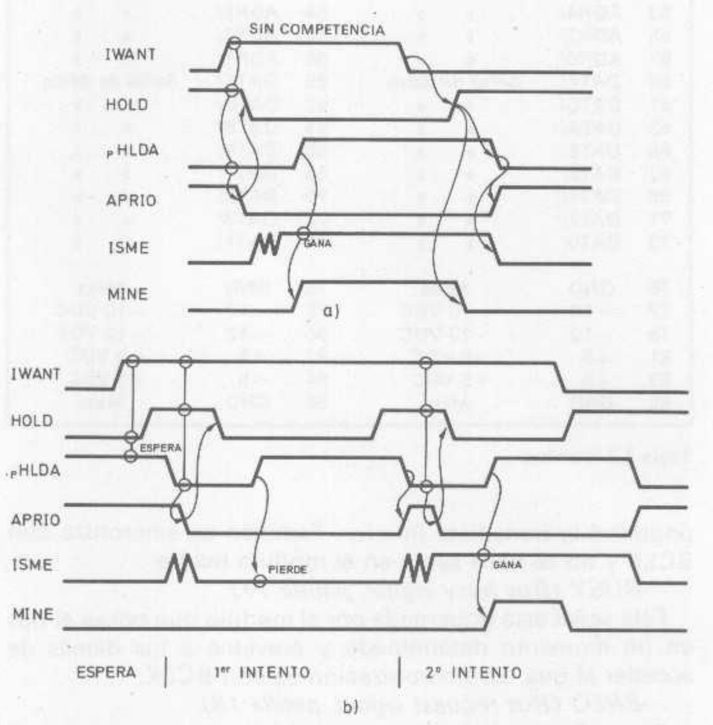


Figura 3.8 a) Cronograma de un arbitraje de bus sin competencia. b) Cronograma de un arbitraje de bus con esperar-perder-ganar donde: IWANT = señal interna que al activarse indica que el amo temporal determina que necesita el bus. APRIO = señal de activación de los bits de prioridad. ISME = señal que resulta del proceso de arbitraje, y que se activa cuando se completa con éxito la comparación de prioridades. MINE = señal de certificación de la toma del bus, resultante de la activación de ISME y del flanco ascendente de pHLDA.

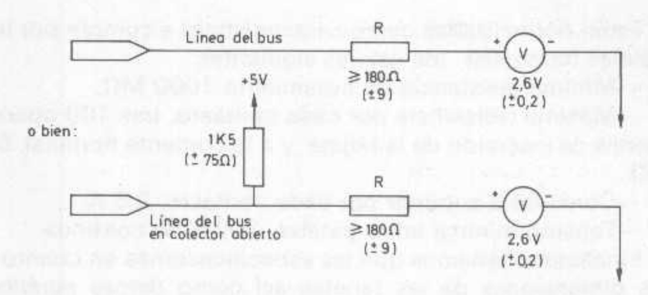


Figura 3.9 Circuitos de terminación de las líneas del bus.

4) Debe activar su prioridad cada vez que actúe la señal HOLD, y debe permanecer activada hasta que llegue el siguiente flanco descendente de la señal pHLDA.

5) Los niveles de prioridad deben ser seleccionables a voluntad mediante conmutadores.

6) Si tras un tiempo de establecimiento suficiente, las únicas líneas con señales en estado activo son aquellas que activó un cierto amo temporal del bus, es éste el que tiene la prioridad más alta y puede tomar el control del bus en cuanto suba la señal pHLDA.

7) Los diseños de cualquier árbitro deben tener en cuenta que las señales del bus y del propio árbitro han de ser estables entre la activación de la señal HOLD y la subida de la señal pHLDA.

3.2.4 Especificaciones eléctricas básicas para las tarjetas S-100

La carga capacitiva total sobre cualquier entrada al bus no debe superar los 25 pF.

Una tarjeta no puede entregar más de 0,5 mA a 0,5 V ni tomar más de 80 μA a 2,4 V, para ninguna de sus líneas excepto las DMA0-3, PHANTOM y PWRFAIL en las que se limita a 0,4 mA a 0,5 V.

Todas las líneas del bus S-100 (excepto las de masa y alimentaciones) deben terminarse, para reducir el ruido, con un circuito equivalente al de la figura 3.9.

3.2.5 Especificaciones mecánicas para los sistemas de interfaz S-100

El conector para las tarjetas será de 100 terminales (doble de 50 más 50). La distancia entre los centros de cada pareja consecutiva de contactos será de 1/8 de pulgada.

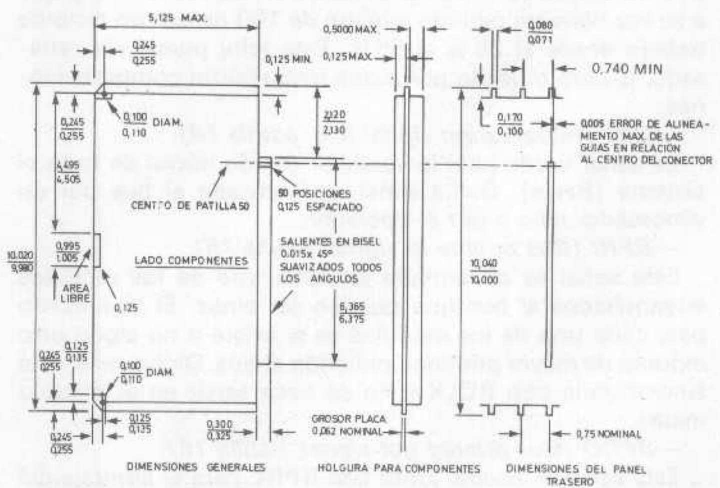


Figura 3.10 Parámetros mecánicos de placas S-100. (Todas las dimensiones están en pulgadas)

Están normalizadas como características a cumplir por las patillas terminales, los valores siguientes:

—Mínima resistencia de aislamiento 1000 MΩ.

—Máxima resistencia por cada contacto, tras 100 operaciones de inserción de la tarjeta, y a la corriente nominal, 50 mΩ.

—Corriente a soportar por cada contacto, 2,5 A.

—Tensión mínima entre patillas, 200 V de continua.

Finalmente tenemos que las especificaciones en cuanto a las dimensiones de las tarjetas así como demás parámetros mecánicos importantes son las mostradas en la figura 3.10.

3.3 MULTIBUS

Es un bus introducido en 1977 por Intel MUL. Pensado primeramente para conectar módulos de Intel, ha sido tal su aceptación que posiblemente constituirá dentro de poco tiempo una norma del IEEE.

Permite la realización de sistemas multiprocesadores de forma que el arbitraje del bus se realiza bajo un mecanismo de encadenamiento (daisy-chain).

En la tabla 3.3 representamos las señales de dicho bus, así como sus nemónicos y una breve descripción de sus significados. Las señales 1-12 y 75-86 inclusive constituyen las alimentaciones, de la 13 a la 34 son las señales de control, de la 35 a la 42 son los vectores de interrupción, las líneas de direcciones ocupan las posiciones 43-58 y las de datos se encuentran en las líneas 59-74. Una barra cruzada de arriba a abajo (/) detrás del nemónico de una señal indica que ésta es activa a nivel bajo.

3.3.1 Descripción de las señales

Para la descripción de las señales las dividiremos tal como lo hemos hecho en la tabla: señales de control del bus, petición de interrupciones, transferencia de datos, las alimentaciones y señales aún no definidas.

3.3.1.1 *Señales de control del bus*

—BCLK (Bus clock, patilla 13)

Es un reloj asíncrono con los de las posibles CPU enganchadas al sistema y cuya misión fundamental es la de sincronizar los circuitos usados para el arbitraje del bus. La sincronización se realiza en el flanco de bajada del reloj, que a su vez tiene un período mínimo de 100 ns con un ciclo de trabajo desde el 35 % al 65 %. Este reloj puede ser retrasado, parado o usado por pasos para realizar comprobaciones.

—INIT (Initialization signal line, patilla 14)

La señal usada para la vuelta al estado inicial de todo el sistema (Reset). Dicha señal será enviada al bus por un procesador amo o por el operador.

—BPRI (Bus priority in signal, patilla 15)

Esta señal es de entrada en cada uno de los módulos enganchados al bus que pueden ser amos. El significado para cada uno de los módulos es si existe o no algún otro módulo de mayor prioridad pidiendo el bus. Dicha señal está sincronizada con BCLK y no se hace servir en el módulo madre.

—BPRO (Bus priority out signal, patilla 16)

Esta señal es usada, junto con BPRI, para el arbitraje del bus. Es una salida de un módulo que puede ser amo del bus y que se une a la BPRI del módulo amo que tenga como prioridad la inmediata inferior. También se sincroniza con BCLK y no se hace servir en el módulo madre.

N.º señal	Nemónico	Descripción	N.º señal	Nemónico	Descripción
1	GND	Masa	2	GND	Masa
3	+5	+5 VDC	4	+5	+5 VDC
5	+5	+5 VDC	6	+5	+5 VDC
7	+12	+12 VDC	8	+12	+12 VDC
9	−5	−5 VDC	10	−5	−5 VDC
11	GND	Masa	12	GND	Masa
13	BCLK/	Reloj del bus	14	INIT/	Puesta del sistema a un estado inicial.
15	BPRI/	Señal de entrada de la prioridad en el bus	16	BPR0/	Señal de salida de la prioridad en el bus.
17	BUSY/	Bus ocupado	18	BREQ/	Petición del bus
19	MRDC/	Orden de lectura en memoria	20	MWTC/	Orden de escritura en memoria.
21	IORC/	Orden de lectura en periférico de Entrada-Salida	22	IOWC/	Orden de escritura en periférico de Entrada-Salida.
23	XACK/	Aceptación de transferencia	24	INH1/	Inhibición de la RAM.
25	AACK/	Aceptación anticipada de transferencia	26	INH2/	Inhibir la ROM o PROM.
27		No usadas	28		No usada
29		No usada	30		No usada
31	CCLK/	Reloj constante	32		No usada
33		No usada	34		No usada
35	INT6/	Petición de Interrupción	36	INT7/	Petición de interrupción
37	INT4/	» »	38	INT5/	» »
39	INT2/	» »	40	INT3/	» »
41	INT0/	» »	42	INT1/	» »
43	ADRE/	Señal de direcciones	44	ADRF/	Señal de direcciones
45	ADRC/	» »	46	ADRD/	» »
47	ADRA/	» »	48	ADRB/	» »
49	ADR8/	» »	50	ADR9/	» »
51	ADR6/	» »	52	ADR7/	» »
53	ADR4/	» »	54	ADR5/	» »
55	ADR2/	» »	56	ADR3/	» »
57	ADR0/	» »	58	ADR1/	» »
59	DATE/	Señal de datos	60	DATF/	Señal de datos
61	DATC/	» »	62	DATD/	» »
63	DATA/	» »	64	DATB/	» »
65	DAT8/	» »	66	DAT9/	» »
67	DAT6/	» »	68	DAT7/	» »
69	DAT4/	» »	70	DAT5/	» »
71	DAT2/	» »	72	DAT3/	» »
73	DAT0/	» »	74	DAT1/	» »
75	GND	Masa	76	GND	Masa
77	−10	−10 VDC	78	−10	−10 VDC
79	−12	−12 VDC	80	−12	−12 VDC
81	+5	+5 VDC	82	+5	+5 VDC
83	+5	+5 VDC	84	+5	+5 VDC
85	GND	Masa	86	GND	Masa

Tabla 3.3 Multibus

—BUSY (Bus busy signal, patilla 17)

Esta señal está gobernada por el módulo que posee el bus en un momento determinado y previene a los demás de acceder al bus. La sincronización es con BCLK.

—BREQ (Bus request signal, patilla 18)

Es usada para indicar que un módulo maestro requiere el uso del bus. La sincronización es con BCLK y no se hace servir en el módulo madre.

—CCLK (Constant clock, patilla 31)

Es un reloj de frecuencia constante para uso general de cualquier módulo. El período mínimo de esta señal es de 100 ns y su ciclo de trabajo de 35 % a 65 %.

3.3.1.2 *Petición de interrupciones*

—INT0-INT7 (Interrupt request lines, patillas 35-42).

Son usadas para petición de las interrupciones. La señal INT0 tiene la prioridad más alta e INT7 la más baja.

3.3.1.3 *Transferencia de datos*

Para realizar las transferencias de datos se usan las líneas de direcciones, las de datos y otras de control para indicar qué tipo de transferencia se quiere realizar y cuándo se termina.

—ADR0-ADRF (Address lines, patillas 43-58)

Son las 16 líneas de direcciones que sirven para direccionar la memoria o registros de entrada-salida. ADR0 contiene el bit de menor peso.

—DAT0-DATF (Data lines, patillas 59-74)

Son 16 líneas bidireccionales usadas para las transferencias de datos entre registros y memoria o viceversa. En los sistemas de 8 bits sólo se usan DAT7-DAT0 siendo DAT0 la señal correspondiente al bit menos significativo.

—MRDC (Memory read command, patilla 19)

Indica que el bus de direcciones contiene una dirección de memoria y que se ha de realizar una operación de lectura. Esta señal es asíncrona con BCLK.

—MWTC (Memory write command, patilla 20)

Indica que el bus de direcciones contiene una dirección de memoria y que el bus de datos contiene el dato que ha de escribirse en dicha dirección de memoria. Esta señal no está sincronizada con BCLK.

—IORC (I/O read command, patilla 21)

Indica que ha de leerse el registro de entrada cuya dirección contiene el bus de direcciones. Esta señal es asíncrona con BCLK.

—IOWC (I/O write command, patilla 22)

Indica que el contenido del bus de datos ha de escribirse en el registro de salida cuya dirección está contenida en el bus de direcciones. Esta señal no está sincronizada con BCLK.

—XACK (Transfer acknowledge signal, patilla 23)

Es una señal generada por el módulo que contiene la posición de memoria o registro de entrada-salida que va a ser accedido para indicar que ya ha sido finalizada la operación de lectura-escritura. Esta señal no está sincronizada con BCLK.

—AACK (Advanced acknowledge signal, patilla 25)

Es una señal que ha sido pensada y puesta en el bus para trabajar con microprocesadores tipo I8080. Esta señal constituye una respuesta rápida a las operaciones de lectura-escritura de forma que el microprocesador pueda realizarlas sin esperar ningún ciclo adicional.

—INH1 (Inhibit RAM signal, patilla 24)

Esta señal permite inhibir a la memoria RAM en los sistemas que contienen en las mismas direcciones memoria RAM y memoria ROM o registros de entrada-salida.

—INH2 (Inhibit ROM signal, patilla 26)

Esta señal permite inhibir a la memoria ROM en los sistemas que tienen en las mismas direcciones memoria ROM y memoria RAM o registros de entrada-salida.

3.3.1.4 *Alimentaciones*

Constituyen las líneas más bajas y más altas del bus. El significado de cada una de las líneas está expresado en la tabla 3.3.

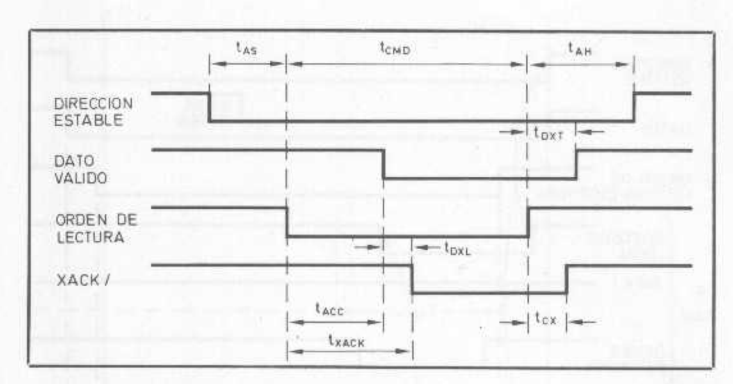


Figura 3.11 Operación de lectura

3.3.1.5 *Señales no definidas*

Para posterior definición y uso se han dejado las patillas número 27, 28, 29, 30, 32, 33 y 34.

3.3.2 Ejemplos de transferencias en el bus

Para finalizar la descripción de MULTIBUS vamos a mostrar los siguientes ejemplos de transferencia por el bus: lectura de un dato, escritura de un dato, una operación de inhibición y un arbitraje del bus con cambio de amo.

3.3.2.1 *Lectura de un dato*

El diagrama de tiempos es el expresado en la figura 3.11. Las direcciones deben estar estables, como mínimo, 50 ns antes que la orden de lectura (t_{AS}). Este tiempo es, aproximadamente, el necesitado por la interfaz para decodificar las direcciones y seleccionar el componente correspondiente.

La orden de lectura debe tener una duración superior a 100 ns y debe terminar, como mínimo, 50 ns antes de modificar la dirección sobre el bus (t_{AH}). Los datos deberán estabilizarse en el bus después de dar la orden y no deberán ser cambiados antes de quitar la orden de lectura. La señal XACK debe coincidir o ser posterior en t_{ACC} a la orden de lectura y en t_{DXL} al instante en que los datos están estables en el bus. Esta señal, XACK, deberá ser quitada del bus un instante t_{CX} posterior a la desaparición de la orden de lectura.

3.3.2.2 *Escritura de un dato*

El diagrama de tiempos es el expresado en la figura 3.12. Como vemos, el dato a ser escrito y la dirección donde ha de escribirse son puestos sobre el bus en el mismo instante. Es decir, t_{DS} tiene el mismo valor que t_{AS}. El intervalo t_{XACK} es el

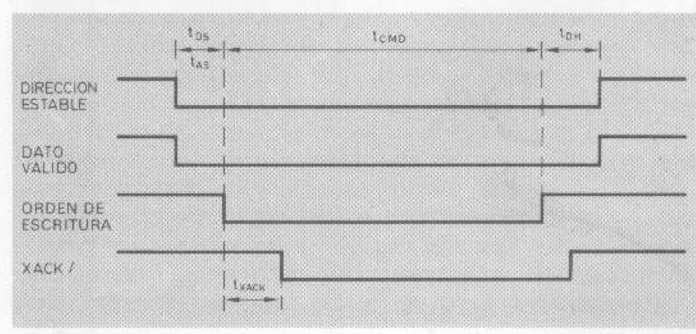


Figura 3.12 Operación de escritura

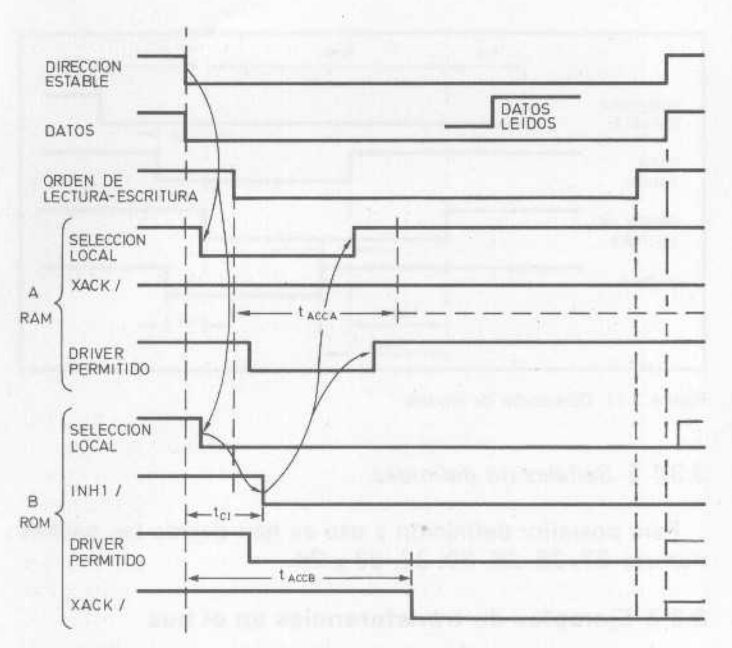


Figura 3.13 Funcionamiento de INH1/

mismo que expresamos en la lectura. Los datos y direcciones deberán permanecer estables durante un intervalo t_{DM} después de cambiar la orden de escritura.

3.3.2.3 *Operación de inhibición*

El uso correcto de las señales INH1 e INH2 permite tener en la misma dirección, memoria RAM, ROM o registros de entrada-salida. En la figura 3.13 hemos puesto los diagramas de tiempo para el caso de tener memoria RAM y PROM en las mismas direcciones y queramos acceder a la PROM. En general se requiere que la señal de inhibición (INH1 en este caso) sea activada lo antes posible (antes de 100 ns, t_{CI}, después de tener dirección estable en el bus) y que la señal X_{ACK} de reconocimiento sea retrasada para permitir al módulo inhibido (en el caso de la figura, la memoria RAM) terminar con cualquier operación iniciada al detectar la orden y una dirección perteneciente a él. Este retraso de X_{ACK} es función del tiempo de acceso de la memoria.

Como vemos en la figura 3.13, después de estabilizarse la dirección son seleccionadas las memorias RAM y PROM. La selección de la PROM produce la señal INH1 que inhibe a la memoria RAM. Como ésta ha sido inhibida después de haber comenzado su ciclo, la señal X_{ACK} de la PROM deberá ser retrasada (t_{ACCB}) hasta después del último X_{ACK} posible de la RAM (t_{ACCA}).

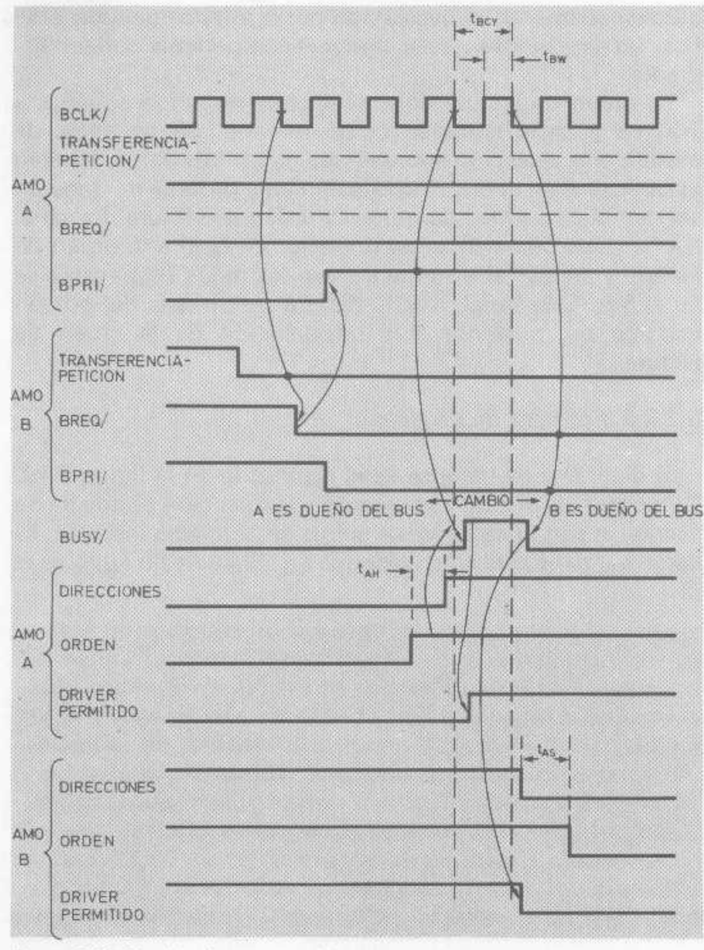


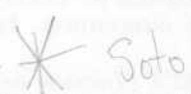
Figura 3.14 Cambio de amo en el bus

3.3.2.4 *Arbitraje del bus*

En la figura 3.14 representamos una secuencia que lleva consigo el cambio de amo del bus. Suponemos que A es un amo con prioridad menor que B junto con que A es el dueño del bus en el momento que B hace una petición (señal transferencia-petición a nivel bajo). Esta señal es sincronizada con el flanco descendente de BCLK para formar BREQ, quien, a su vez, hace subir la señal BPRI del amo A, indicándole que existe una petición de bus de un posible amo más prioritario que él. Después de que A realiza la transferencia en curso se realiza el cambio de dueño del bus. Esta función se realiza, tal como vemos en la figura, en un ciclo de reloj BCLK.

En la tabla 3.4 se representan los valores de los tiempos de las señales empleadas en los diagramas de tiempo anteriores.

3.4 EL SISTEMA CAMAC



La norma para Control y Medida Automatizados por Ordenador, (Computer Automated Measurement and Control= CAMAC), fue desarrollada inicialmente en 1969 por el European Standards on Nuclear Electronics. Tres años más tarde los informes 4100 y 4600 del Euratom definieron la interconexión y las señales, temporización y organización lógica de las uniones entre controladores.

El IEEE ha adoptado CAMAC como norma para el área de la interconexión digital mediante el documento principal IEEE Std. 583 — 1975 y otros relacionados con él. Actualmente más de 75 empresas ofrecen productos CAMAC.

3.4.1 Generalidades

CAMAC es un estándar que cubre todas las áreas relativas a la interconexión de instrumentos, mediante un concepto modular de la instrumentación para asegurar la unión entre transductores y conjuntos de dispositivos para el tratamiento digital de la información, (ordenadores, periféricos, controladores externos...).

El concepto modular de CAMAC se enfoca hacia la obtención de las ventajas siguientes:

—posibilitar la realización rápida de modificaciones en sistemas que presentan nuevos requerimientos;

Parám.	Mín.	Máx.	Descripción	Comentarios
t_{BCY}	100 ns		Anchura del pulso del reloj del bus	
t_{BW}	$0{,}35xt_{BCY}$	$0{,}65xt_{BCY}$	Anchura del reloj bus	
t_{AS}	50 ns		Tiempo de establecimiento de las direcciones	Relativo a la activación de la orden
t_{DS}	50 ns		Tiempo de establecimiento de los datos a escribir	Relativo a la activación de la orden
t_{AH}	50 ns		Tiempo de mantenimiento de las direcciones	Relativo al cambio de la orden
t_{DH}	50 ns		Tiempo de mantenimiento de los datos a escribir	Relativo al cambio de la orden
t_{DXL}	0 ns		Tiempo de establecimiento de los datos leídos	Relativo a la aceptación (XACK/)
t_{DXT}	0 ns		Tiempo de mantenimiento de los datos leídos	Relativo al cambio de la orden
t_{CX}	0 ns	100 ns	Tiempo de mantenimiento de la aceptación	Relativo al cambio de la orden
t_{XACK}	0 ns	100 ms	Retraso de la aceptación	
t_{ACC}	0 ns	$t_{XACK}-t_{DXL}$	Tiempo de acceso de la lectura	
t_{CMD}	100 ns		Anchura del pulso de la orden	
t_{CI}		100 ns	Retraso de la inhibición	Relativo a la dirección
t_{ACCB}	Depende del tiempo de ciclo del esclavo		Aceptación del esclavo que inhibe a otro	

Tabla 3.4

—posibilitar el test independiente de diversos módulos funcionales antes de que lleguen a formar parte de un sistema complejo, para ahorrar de este modo tiempo de comprobación global;

—el intercambio rápido de los módulos de mantenimiento, con lo que se logra disminuir el tiempo de no disponibilidad.

Las especificaciones de CAMAC son muy rigurosas (hay que contar con el hecho de que originariamente estas normas eran de aplicación exclusiva en la industria nuclear) y, por lo tanto, los equipos CAMAC han de cumplir valores muy exactos en todos sus parámetros tanto mecánicos como eléctricos.

Las características de este sistema no son las más apropiadas, en nuestra opinión, para aplicaciones de control o de proceso de datos. Para el caso concreto de los microcomputadores, aunque un sistema como CAMAC presenta la ventaja de poseer un buen soporte para el usuario, tiene el grave inconveniente de que resulta probablemente demasiado costoso (al menos en su forma de bus paralelo) para las interfaces usuales. Teniendo en cuenta esta importante limitación, se va a presentar aquí este sistema sin mucho detalle, dejando para el lector específicamente interesado en CAMAC el examen más a fondo de todos los documentos que especifican las características de esta norma y cuya referencia se incluye al final del tema.

3.4.2 Definiciones

Se hace preciso, antes de presentar las principales características de CAMAC, definir una serie de términos que son particulares de este sistema.

La subunidad básica se llama un «crate», a lo que aquí llamaremos genéricamente un módulo.

Se definen aquí dos vías de comunicación: el bus de datos (Dataway), que sirve para interconexiones dentro de un módulo, y el bus principal (Highway), que permite las transferencias entre módulos y que está definido tanto en su versión como camino serie que como camino paralelo.

Cada módulo va provisto de un elemento esencial que es el controlador (crate controller). Queda así configurada una interfaz en la que se ha normalizado cuidadosamente el módulo unitario completo, pero que permite flexibilidad en cuanto a la concreta función interna que realice, que puede ser, por ejemplo, una operación de control y (o) medida de un proceso físico exterior al sistema.

En cada módulo pueden encontrarse hasta 24 dispositivos individuales que se llaman estaciones, una de las cuales es el controlador.

3.4.3 Especificaciones generales para una interfaz CAMAC

Cuando se trata de resolver un problema de conexión de unos ordenadores a unos procesos en tiempo real, es muy útil disponer de un medio tan eficiente como lo es una interfaz estándar diseñada y construida con independencia de cual sea la puerta o bus de E/S del ordenador.

Pues bien, en el núcleo de la norma CAMAC lo que hay es una estructura normalizada de bus que responde al esquema de la figura 3.15.

Todos los módulos del esquema pueden ser funcionalmente independientes del ordenador central y abarcar una amplia gama de unidades compatibles CAMAC, que actualmente comprende como ejemplos comercialmente disponibles (entre otros):

—módulos de uso general, como contadores, temporizadores, registros serie y paralelo, generadores de reloj, impulsos y palabras, visualizadores;

—multiplexores y conversores;

—interfaces para periféricos tales como lectores de cinta de papel y de tarjetas, impresoras de líneas, controladores de cassette y teletipo, visualizadores de pantalla, trazadores de gráficos;

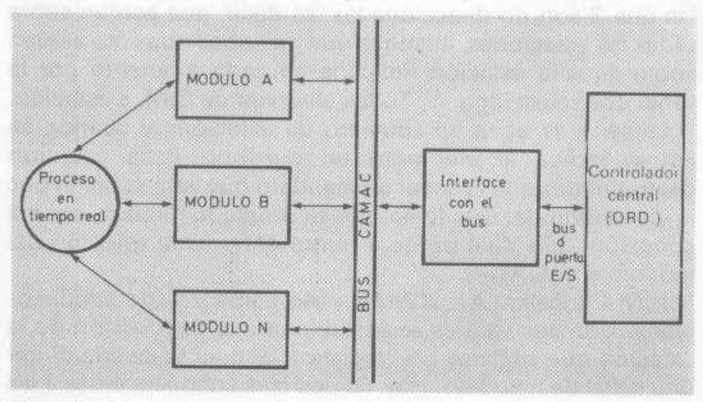


Figura 3.15 Estructura general de interfaz estándar en CAMAC.

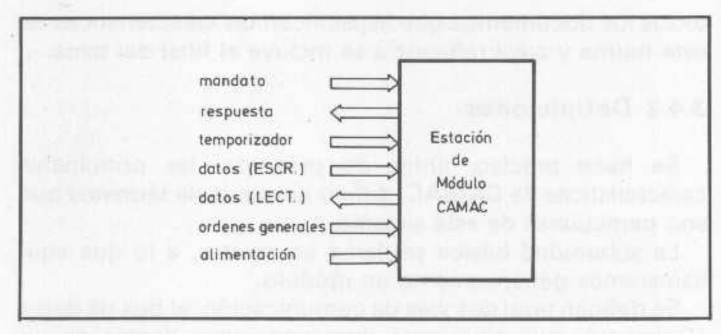


Figura 3.16 Interface básico con una estación CAMAC.

—controladores de módulo;

—circuitos excitadores paralelos para buen número de modelos de minicomputadores (PDP—8, 9, 11, 15; HP 2100; Nova; Honeywell 316/516; Siemens 320/330, 404/3; SPC-16, etc.);

—módulos (crates) y circuitería asociada incluyendo fuentes de alimentación.

Básicamente la interfaz con el bus de datos es como se muestra en la figura 3.16, donde todas las señales se normalizan como niveles TTL.

Para la interconexión de un módulo con el ordenador exterior (o con otros módulos o con un controlador) se utiliza un elemento especializado que siempre se encuentra insertado en la parte derecha de cada módulo según la norma CAMAC, y que es el controlador de módulo (C.M.). Este controlador puede muy bien ser un microcomputador, en cuyo caso no se necesita añadir inteligencia a la interfaz. En todo caso cualquier C.M. que se ajuste a la norma IEEE 583-75 permite la intercambiabilidad de módulos, ya que la conexión con el bus de datos está perfectamente normalizada.

En resumen puede afirmarse que CAMAC es un sistema normalizado en el que la conexión al bus es de poca complicación lógica, pero en el que lo complejo es la propia unidad básica. Su uso en una aplicación pequeña puede resultar excesivamente costoso, mientras que efectuar ampliaciones sobre diseños bastante complicados resulta sencillo y de coste relativamente bajo. La aceptación de CAMAC no parece muy clara para aplicaciones con μC.

3.4.3.1 *Bus de datos (Dataway) en CAMAC*

Se compone de 86 líneas que están disponibles para cada una de las estaciones normales (todas salvo el C.M.) de un módulo. Este bus permite transferencias a velocidad máxima de 24 Mbits/s.

1. Hay un conjunto de 13 señales de control y órdenes, de las que 3 son no direccionadas, es decir, que actúan sobre todas las estaciones, mientras que hay otras que sólo actúan sobre aquella estación indicada en cada momento por la señal que selecciona, *N*. Todas ellas van de C.M. a estación.

La señal *N* es la de «número de estación» y cuando se activa indica al elemento en cuestión (una estación determinada) que la señal de mandato que está sobre el bus le viene dirigida. Esta línea, por su propia naturaleza, es una conexión individual punto a punto entre cada estación del módulo y el C.M.

Hay 4 señales, *A1, A2, A4* y *A8* llamadas «de subdirección». Con sus valores se selecciona una subdivisión de la estación que se llama un «registro», al cual se puede dirigir una señal de mandato. Hay 16 posibles subdirecciones a las que dirigir las señales *M*.

Tipo de las señales	*nombre simbólico*	*n.º de contactos*	*Función de las señales*
	N	1	Selecciona una estación dentro del módulo.
ORD.	A1-2-4-8	4	Seleccionan posiciones dentro de la estación.
(direcc.)	F1-2-4-8-16	5	Definen la operación a realizar durante los mandatos.
Secuenciam.	S1 y S2	2	Señales de validación.
DATOS	W1 a W24	24	Bus de escritura.
	R1 a R24 (1)	24	Bus de lectura
	L (1)	1	Indica petición de interrupción.
ESTADO	B	1	Indica bus ocupado
	Q (1)	1	Respuesta a ciertos mandatos.
	X (1)	1	Aceptación de mandato, indica la capacidad de una estación para ejecutar la operación.
CONTR. GENER.	Z	1	Inicializar.
	I	1	Inhibir.
	C	1	Poner a cero.
NO ESPECIFICADAS.	—	5	Sin definir.
ALIMENCION	24 V	1	
	6 V	1	
	− 24 V	1	
	− 6 V	1	
	retorno	2	
ADICIONALES DE ALIM.	varias tensiones	8	

(1) señales que van desde estación normal de módulo hacia el controlador de módulo (C.M.). Las restantes son en sentido contrario.

Tabla 3.5 Bus de datos («Dataway») CAMAC

Para el envío de estas señales de mandato *M* se tienen 5 líneas de función, *F1, F2, F4, F8* y *F16*. Mediante sus valores binarios se obtiene un código que permite definir 32 posibles funciones, de las cuales no todas están especificadas actualmente por la norma CAMAC. Por ejemplo, F(0), A(1) significa leer datos desde el «registro» 1; F(16) significa escribir datos hacia el «registro» que se especifique; F(25) vale por «ejecutar» lo que puede significar la salida de un cierto impulso; etc.

Las señales que actúan como mandatos para todas las estaciones de un módulo son *Z* (Initialize), *C* (Clear) e *I* (Inhibit). Son órdenes generales que envía el C.M. a cualquier estación.

Z es la señal para pasar los registros a un estado inicial, en especial cuando se conecta la alimentación.

C es la señal para borrar contenidos de registros, o limpiar básculas biestables.

I permite inhabilitar unas características determinadas durante la duración de la señal.

Siempre que actúe una señal de control de cualquier tipo, el C.M. envía una señal sobre la línea *B* que es de estado y sirve para indicar la ocupación del bus.

2. Existen 2 señales específicas generadas por el C.M. para secuenciamiento que son las *S1* y *S2* (strobes). Actúan durante las operaciones de mandato en el ciclo de bus. Durante las operaciones *Z, C* e *I* sólo es obligatoria la *S1*.

La señal *S1* es de validación para que las estaciones acepten datos del bus; y la señal *S2* es otra de validación

para que las estaciones inicien operaciones que puedan cambiar el estado de las líneas de lectura o escritura del bus de datos.

El ciclo de bus CAMAC resulta así una secuencia síncrona de 2 fases en que los datos se transfieren (leen o cargan) con *S1*, y se cambian (borran o secuencian) con *S2*.

No hay un tiempo de ciclo estándar, pero se especifica un mínimo de 1 μs.

3. Líneas para datos hay 48, la mitad para lectura y la mitad para escritura. Debe advertirse que aquí «datos» es un término general que puede incluir también a las informaciones de control, y que lectura y escritura son términos que sólo implican cuál es el sentido de la transferencia.

Todos los datos se comunican en paralelo dentro de un módulo. Por el bus de lectura R1-R24 se transmite información digital desde una estación hacia el C.M. según un formato de palabras de 24 bits como máximo (tres octetos simultáneos). Por el bus de escritura W1-W24 idéntica operación pero en sentido inverso. Puede pensarse que en sistemas microcomputadores estas 24 líneas podrían llevar las direcciones y los datos, con lo que el ancho de banda del bus se elevaría.

4. De las 4 líneas de estado que hay en este bus, una de ellas (*L*) tiene en común con la línea *N* el ser una conexión individual punto a punto entre cada estación del módulo y el C.M. Son éstas las dos únicas líneas que no enlazan entre sí a todos los elementos individuales del módulo.

La señal *L* (look at me) —Mírame— se activa para indicar al C.M. que una estación pide atención o servicio. El uso de las líneas individuales *N* y *L* permite acciones simultáneas en varias estaciones.

Hay especificados dos bits de respuesta, que retornan desde cada estación al C.M. a través de las líneas X y Q.

La señal *X* es la de «mandato aceptado» y *Q* es una señal de respuesta generalizada que el diseñador puede acomodar a sus necesidades; esta señal se usa también para transferencias de bloque de las que están normalizadas tres clases diferentes. La activación de la señal *X* por una estación indica que tiene capacidad para ejecutar la operación especificada por la orden en curso.

La señal *B* (busy) —ocupado— indica que está avanzando la realización de una cierta operación sobre el bus. Actúa sobre la totalidad de las estaciones.

5. Además de las 67 líneas anteriores, la norma CAMAC incluye en el bus de datos una serie de líneas para alimentación de las estaciones.

Las tensiones estándar son +6 V, —6 V, +24 V, y —24 V. Hay además terminales para alimentaciones adicionales de uso en casos especiales (12 V, −12 V, 117 V de alterna, 200 V) y terminales para la masa del sistema. Todas ellas con unas especificaciones muy rigurosas sobre estabilidad, regulación y supresión de transitorios.

Por último hay 5 líneas que permanecen sin asignación.

3.4.3.2 *Otras especificaciones*

CAMAC consta de una serie de documentos que describen varias organizaciones de sistemas y (o) niveles de compatibilidad. Así, la norma IEEE—583—1975 se ocupa, como ya hemos visto, de la pareja módulo/estación. La norma IEEE 596 de la interconexión rápida de hasta 7 módulos o sea que describe un «bus principal» con 66 pares de líneas. La norma IEEE-595 trata de un «bus principal» serie para aplicaciones más lentas que requieran transferencias de datos a larga distancia a nivel de bit o de octeto en serie, entre un máximo de 62 módulos, y basándose en un bucle unidireccional; en particular especifica normas para señales según la EIA RS-422, pero es posible tener otras formas de comunicación serie entre módulos usando adaptadores. La interconexión serie puede ser de especial interés para usuarios industriales en el campo del control de procesos.

Otro elemento útil de CAMAC es una estandarización software que proporciona un lenguaje común para estos sistemas que es el IML.

3.4.3.3 *Algunas aplicaciones típicas*

Todavía en la actualidad hay que admitir que la mayoría de las interfaces CAMAC se aplican en la automatización de laboratorios de física nuclear y física de alta energía. Existen, no obstante, algunas otras aplicaciones en sistemas médicos, industriales y de medida. Vamos a citar varios ejemplos.

En el campo del control de procesos industriales podemos señalar que la Aluminum Co. of America instaló en 1976 un gran sistema CAMAC formado por 23 módulos que controlan 45 hornos y que instrumenta un «bus principal» serie con bucle de reserva que opera con cable coaxial a velocidades de 500 Kbaudios.

Otro campo de aplicación ha sido el de adquisición de datos y el de la interconexión de periféricos estándar y ordenadores o estos últimos entre sí. En la Universidad de Stanford (Calif. — EE.UU.) se ha realizado un pequeño sistema uniusuario de adquisición de datos, interconectando periféricos a un miniordenador PDP 11-40 sobre el Unibus de DEC. La mayoría de los datos se tomaban a través de estaciones CAMAC comerciales, y se utiliza un «bus principal» paralelo normalizado.

Un área de creciente interés donde puede aplicarse CAMAC es el de los enlaces en una red de ordenadores; existe un sistema desarrollado en el Daresbury Lab., en Gran Bretaña, donde un IBM 370/165 hace de nodo central y una serie de miniordenadores supervisan las adquisiciones de datos experimentales, mientras otro ordenador de control se encarga de una serie de medidas con aparatos específicos, y existen un conjunto de enlaces con varias estaciones remotas.

3.4.4 Características mecánicas

Los equipos CAMAC se ensamblan instalando en un chasis normalizado una serie de «cajones», cada uno de los cuales lleva un conector de 86 contactos y constituye una o varias estaciones del módulo que en total pueden sumar desde 1 hasta 24. Mediante aquellos conectores cada estación puede acceder al bus de intercambio de informaciones.

Los chasis tienen una altura de 19 pulgadas y el espaciado entre estaciones es de 17,2 mm, siendo la longitud variable, dependiendo del número de estaciones. La última de éstas a la derecha, visto el sistema desde el frente, es la que actúa como estación de control (C.M.) y es la única que no puede acceder a las líneas de datos. Existen unas líneas específicas que unen un contacto de cada estación normal con un contacto del C.M.

Con objeto de alcanzar una intercambiabilidad completa a nivel mecánico, CAMAC no empaqueta de forma independiente el equipo que hay a cada lado de la frontera donde se define la interfaz, sino que especifica la normalización no sólo de los atributos físicos de cables y conectores, sino también los de cada equipo como unidad modular; esto incluye desde estandarizar los armarios (racks) hasta

especificar completamente las dimensiones de todas las piezas que componen el equipo.

3.4.5 Perspectivas de CAMAC y reseñas informativas

Las últimas actividades que sobre esta norma se están llevando a cabo van en las líneas de trabajo siguientes:

—limitación del número de modos diferentes de transferencia de bloques para simplificar los controladores y las interfaces, y para ayudar a que el diseñador de módulos seleccione el mejor modo de operación;

—estudio de la relación de la norma CAMAC con otras normas digitales, en particular con la IEEE-488 para instrumentos programables;

—estudio de la influencia de los microprocesadores sobre CAMAC con objeto de poder aprovechar al máximo todo el potencial de control distribuido que posee la estructura CAMAC;

—construcción de subrutinas FORTRAN usando la semántica del lenguaje IML;

—extensión del BASIC para abarcar funciones de CAMAC.

Para conseguir una información más detallada sobre esta norma hay diversas fuentes generales, como son la revista cuatrimestral «Camac Bulletin» de la «Commission des Communautés Européennes», y diversas instituciones de EE.UU. como el «National Bureau of Standards» y en particular el «IEEE Camac Standard Committee», que ha publicado en fecha reciente un volumen que incluye toda la información de interés para usuarios de sistemas CAMAC que hasta el momento ha sido estandarizada.

3.5 NORMA PARA LA INTERCONEXIÓN DIGITAL DE INSTRUMENTOS PROGRAMABLES: IEEE-488 — (ANSI-MC. 1.1.—1975)

Este bus se utiliza fundamentalmente en la interconexión entre uno o varios procesadores y periféricos inteligentes. El hecho de que el protocolo de la transmisión sea asíncrono hace que el bus pueda extenderse hasta 20 m, con cargas cada 2 m. Su uso se extiende cada vez más, existiendo interfaces para toda clase de ordenadores, incluso los personales como el PET de Comodore.

El bus debe contar con un dispositivo que actuará de controlador, encargado de gestionar las peticiones de uso del bus, ya sea para emitir mensajes (locutor), ya para recibirlos (oyente).

Intel y Motorola han puesto a la venta sendas pastillas que realizan las funciones de control de la transferencia de datos en el bus (Locutor y Oyente [L, LE, T, TE]).

Intel, además, ha programado una pastilla UPI-41 para que se comporte como controlador del bus (C).

Esta norma, aprobada por el IEEE en 1975, fue ideada por Hewlett-Packard para interconectar sus aparatos de medida programables. Hewlett-Packard tiene patentadas en USA las especificaciones eléctricas y funcionales del bus, pero ha garantizado al IEEE que cederá la licencia de utilización a quien lo solicite, en términos razonables.

Esta descripción del bus no pretende ser exhaustiva, pues ello requeriría una extensión mayor que la del capítulo completo. Con ella sólo se pretende aclarar los conceptos necesarios para comprender los fundamentos del funcionamiento del bus, de forma que el lector pueda ponderar las ventajas e inconvenientes del mismo y, caso de decidir la conveniencia de utilizarlo en una aplicación concreta, pueda asimilar con facilidad una descripción detallada del mismo.

El IEEE ha publicado un folleto con las especificaciones completas de la norma. Motorola también ha publicado un folleto en el que se detallan las especificaciones del bus, como nota de aplicación de la pastilla a que nos hemos referido antes.

3.5.1 Señales eléctricas del bus

El bus consta de 16 señales activas, divididas en tres grupos:

— Bus de datos (8 líneas).

— Bus de control de transferencia de datos (3 líneas).

— Bus para el control general de la interconexión (5 líneas).

3.5.1.1 *Bus de datos*

Un conjunto de ocho líneas de datos permiten trasmitir por el bus, de octeto en octeto. La denominación de las líneas es:

DIO1 (Data Input Output 1): Dato de entrada o salida 1

DIO8 (Data Input Output 8): Dato de entrada o salida 8

3.5.1.2 *Bus de control de transferencia*

Consta de tres señales, usadas para efectuar la transferencia de cada octeto entre el locutor y el oyente a través de las señales DIOn. La denominación y descripción de estas señales es la siguiente:

—*DAV (Data Valid):* Dato válido. Es emitida por el locutor de la transferencia, e indica que los datos DIOn están estables en el bus.

—*NRFD (Not Ready For Data):* No listo para dato. Es emitida por el oyente de la transferencia, e indica que aún no está listo para recibir nuevos datos.

—*NDAC (Not data accepted):* Dato no aceptado. Es emitida por el oyente de la transferencia, e indica al locutor que debe mantener los datos en el bus, porque aún no han sido almacenados.

3.5.1.3 *Bus de control general de la interconexión*

Comprende cinco señales, que son empleadas para mantener un flujo ordenado de información a través del bus. Son las siguientes:

—*ATN (Attention):* Atención. Es empleada por el dispositivo que hace las funciones de controlador del bus e indica a todas las demás que está enviando un mensaje de interés general.

—*IFC (Interface clear):* Limpiar la interconexión. El controlador indica al resto de los dispositivos que deben volver al estado inicial (Reset), o de reposo.

—*SRQ (Service Request):* Petición de servicio. Los dispositivos no controladores usan esta línea para indicar al controlador sus deseos de utilizar el bus para efectuar una transferencia de datos.

—*REN (Remote Enable):* Válida control remoto. El controlador indica a los dispositivos direccionados que deben ignorar el control local, panel frontal o similar, para obedecer al control remoto recibido a través del bus.

—*EOI (End or Identify):* Fin o identificación. Puede ser

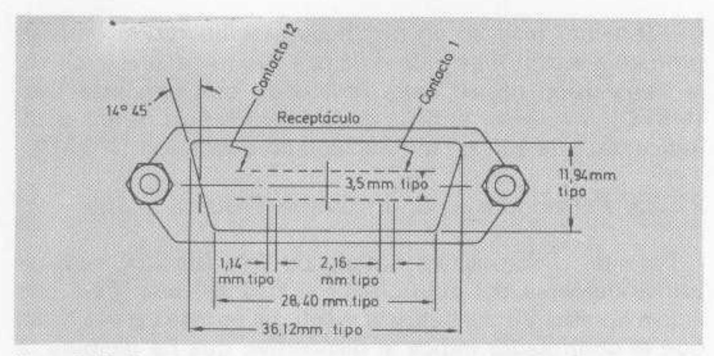


Figura 3.17 Croquis del conector normalizado del bus IEEE-488. Corresponde a los tipos MICRORIBBON (Serie 57 de Amphenol o Cinch) o CHAMP (AMP).

activada por el dispositivo locutor o por el controlador. En el primer caso indica el fin de la transmisión de un bloque de datos. En el segundo, el controlador indica a los dispositivos que han pedido servicio que se identifiquen.

3.5.2 Especificaciones mecánicas

El tipo de conector recomendado en la norma del IEEE es el de la figura 3.17. Es un conector trapezoidal de 24 contactos, en los que las señales se distribuyen de la forma indicada en la tabla 3.6.

Todas las señales se transmiten por el bus con lógica negativa (1 = Bajo $\leqslant$ 0,8 V; 0 = Alto $\geqslant$ 2,0 V).

3.5.3 Funciones a realizar por los dispositivos conectables al bus

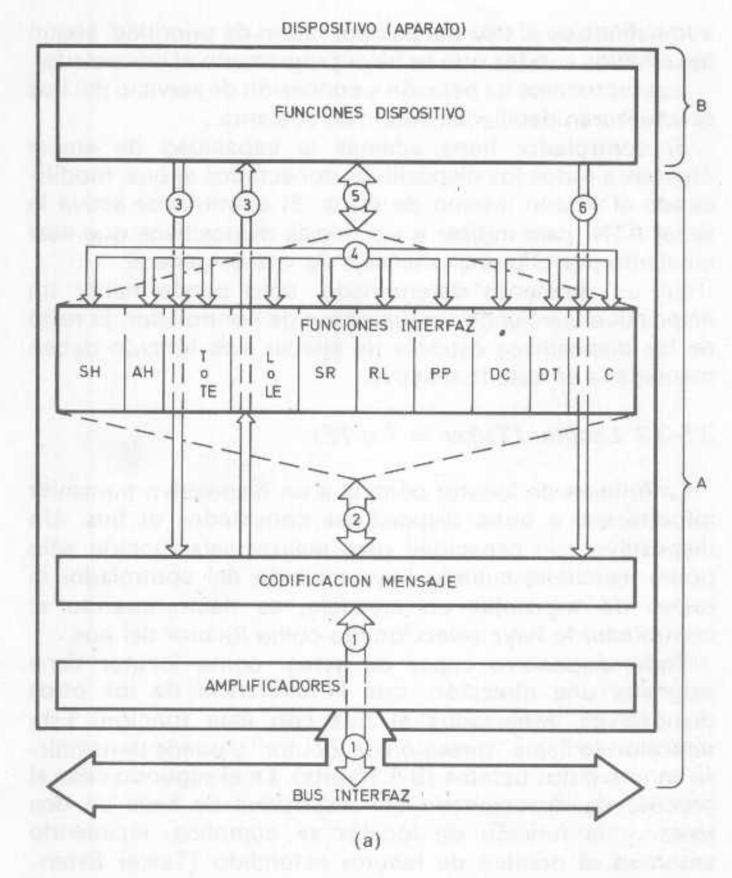
Para que un dispositivo sea capaz de interpretar una señal o conjunto de ellas es preciso que se le haya dotado de los circuitos necesarios para analizarlos y responder en función de dichas señales. Estos circuitos activarán o no unas señales, locales del dispositivo o generales de bus, en función de las señales que interpreten.

Según esto, un dispositivo se considera descompuesto en diez funciones distintas, cada una de las cuales es relativamente independiente de las demás y se puede analizar separadamente. El análisis detallado de cada una de estas funciones requeriría la transcripción literal de la información contenida en los folletos del IEEE y Motorola, lo cual nos apartaría del objeto de estas líneas, que es ofrecer al lector una visión lo más clara y resumida posible de las posibilidades del bus.

Las funciones que puede realizar un dispositivo conectado al bus son las indicadas en la figura 3.18.

Contacto	*Señal de la línea*	*Contacto*	*Señal de la línea*
1	DIO 1	13	DIO 5
2	DIO 2	14	DIO 6
3	DIO 3	15	DIO 7
4	DIO 4	16	DIO 8
5	EOI	17	REN
6	DAV	18	Gnd, (6)
7	NRFD	19	Gnd, (7)
8	NDAC	20	Gnd, (8)
9	IFC	21	Gnd, (9)
10	SRQ	22	Gnd, (10)
11	ATN	23	Gnd, (11)
12	SHIELD	24	Gnd, LOGIC

Tabla 3.6 Asignación de las líneas del bus a los terminales del conector Gnd (*n*) se refiere a la señal de retorno del contacto *n*.



Función del interfaz	*Símbolo*	*Caminos empleados por la función*
Protocolo surtidor	SH	1, 2, 4, 5
Protocolo receptor	AH	1, 2, 4, 5
Locutor o locutor extendido	T o TE	1, 2, 3, 4, 5
Oyente u oyente extendido	L ó LE	1, 2, 3, 4, 5
Petición de servicio	SR	1, 2, 4, 5
Control remoto o local	RL	1, 2, 4, 5
Encuesta paralelo	PP	1, 2, 4, 5
Borrado del dispositivo	DC	1, 2, 4, 5
Disparo del dispositivo	DT	1, 2, 4, 5
Controlador	C	1, 2, 4, 5, 6

(b)

Figura 3.18 Descomposición funcional de un dispositivo conectado al bus IEEE-488. a) Descomposición en bloques del dispositivo completo. A= Capacidad definidad por esta norma. B = Capacidad definida por el diseñador. 1 = Líneas del bus. 2 = Mensajes remotos del interfaz, de y hacia las funciones del interfaz. 3 = Mensajes dependientes del dispositivo, de y hacia las funciones del dispositivo. 4 = Relaciones entre los estados de las funciones del interfaz. 5 = Mensajes locales entre las funciones del dispositivo y las funciones del interfaz. (Los mensajes hacia las funciones del interfaz están definidos, los mensajes desde las funciones del interfaz dependen de las elecciones hechas por el diseñador). 6 = Mensajes remotos del interfaz enviados por las funciones del dispositivo que actúa como controlador. b) Repertorio de funciones del interfaz.

3.5.3.1 *Controlador (Controller = C)*

El dispositivo que realiza las funciones de controlador es aquél encargado de inicializar todas las comunicaciones, que se establezcan a través del bus.

En el bus debe haber siempre un dispositivo que reciba las peticiones de uso del bus, ya sean de un dispositivo que desea emitir un mensaje, ya sea de uno que espera recibirlo. Este dispositivo analizará las peticiones de servicio, averiguando el dispositivo o dispositivos que las han realizado y

autorizándoles al uso del bus por orden de prioridad, según los criterios con los que se haya programado el controlador.

Los protocolos de petición y concesión de servicio del bus se analizarán detalladamente más adelante.

El controlador tiene además la capacidad de enviar órdenes a todos los dispositivos conectados al bus, modificando el estado interno de éstos. El controlador activa la señal ATN, para indicar a los demás dispositivos que está enviando por el bus un mensaje de interés general.

En un momento determinado, sólo puede haber un dispositivo ejerciendo las funciones de controlador. El resto de los dispositivos capaces de ejercer esta función deben mantenerla en estado inactivo.

3.5.3.2 *Locutor (Talker = T o TE)*

La función de locutor permite a un dispositivo transmitir información a otros dispositivos conectados al bus. Un dispositivo con capacidad para realizar esta función sólo podrá ejercitarla cuando haya recibido del controlador la orden de transmitir un mensaje, es decir, cuando el controlador lo haya seleccionado como locutor del bus.

Todo dispositivo capaz de actuar como locutor tiene asignada una dirección, que le diferencia de los otros dispositivos conectados al bus con esta función. Esta dirección se llama "dirección de locutor" y puede transmitirse en uno o dos octetos (5 ó 10 bits). En el segundo caso el proceso de direccionado del dispositivo se hace en dos fases, y la función de locutor se complica, recibiendo entonces el nombre de locutor extendido (Talker Extended = TE).

3.5.3.3 *Oyente (Listener = L o LE)*

La función de oyente capacita al dispositivo que la ejercita para recibir mensajes a través del bus. Estos mensajes serán transmitidos por el dispositivo que esté ejerciendo en ese momento la función de locutor.

La función de oyente sólo puede ser ejercida por los dispositivos, que han sido seleccionados por el controlador como oyentes del bus. Lo mismo que en el caso del locutor, esta selección la hace el controlador enviando por el bus de datos la "dirección del oyente", que puede transmitirse en uno o dos octetos. En el segundo caso la función que permite al dispositivo reconocerse como oyente del bus se llama "oyente extendido" (Listener extended = LE).

3.5.3.4 *Protocolo surtidor (Source Handshake = SH)*

Esta función (SH) permite al dispositivo enviar un dato cualquiera por el bus, ya sea parte de un mensaje, ya una respuesta a una orden del controlador. La función de locutor (T) solamente capacita al dispositivo para adquirir el control del bus de datos, como dispositivo emisor de mensajes. Se requiere esta función (SH) para controlar el protocolo de las señales, que asegura la llegada del dato transmitido a todos los dispositivos que han sido seleccionados para aceptarlo.

El protocolo de la transmisión de datos es asíncrono, y lo analizaremos detalladamente más adelante.

3.5.3.5 *Protocolo receptor (Acceptor Handshake = AH)*

Esta función (AH) permite al dispositivo recibir los datos transmitidos por el bus. Esta función está directamente relacionada con el protocolo del surtidor del mensaje (SH).

Dado que la trasferencia de datos es asíncrona, y que puede haber más de un dispositivo recibiendo el dato, el protocolo receptor permite al dispositivo retardar el inicio de la transmisión de un dato, indicando que uno está listo (NRFD), y retardar el final de la transferencia de un dato, indicando que aún no lo ha aceptado o recibido (NDAC).

3.5.3.6 *Petición de servicio (Service Request = SR)*

Permite, al dispositivo dotado de esta función, solicitar asíncronamente del controlador el uso del bus. Para ello activa la señal SR del bus, y la mantiene en este estado hasta que el controlador indica al dispositivo que ha recibido la solicitud de servicio. Más adelante se detallará el protocolo concreto del diálogo entre el solicitante de servicio y el bus.

3.5.3.7 *Encuesta en paralelo (Parallel Poll = PP)*

Esta función proporciona al dispositivo la capacidad de presentar un bit de estado al controlador sin haber sido direccionado como locutor.

Esta capacidad es ejercida por el dispositivo cuando el controlador hace una encuesta en paralelo, preguntando de esta forma a una serie de dispositivos preseleccionados si requieren o no el servicio de bus. Ello permite al controlador discernir entre ocho dispositivos, o grupos de dispositivos, uno por cada una de las líneas de datos (DIO) del bus, con una sola pregunta.

3.5.3.8 *Borrado del dispositivo (Device Clear = DC)*

Esta función permite borrar (inicializar) el dispositivo, individualmente o como parte de un grupo de ellos. El grupo puede ser un subconjunto o la totalidad de los dispositivos direccionables conectados al bus.

Es necesario distinguir entre esta función DC y la señal IFC del bus. La función DC inicializa el dispositivo, llevándolo al estado inicial para el que ha sido programado. La señal IFC, en cambio, inicializa las funciones de interconexión con el bus y no afecta al dispositivo conectado al bus a través de estas funciones.

3.5.3.9 *Disparo del dispositivo (Device Trigger = DT)*

Esta función permite poner en marcha las funciones básicas del dispositivo, individualmente o en grupo, lo mismo que en la función DC.

3.5.3.10 *Control remoto o local (Remote Local = RL)*

Permite al dispositivo seleccionar entre dos fuentes de información de entrada. La función indica al dispositivo si debe usar la información recibida del panel frontal, local, o la recibida a través de la interconexión al bus, remota.

3.5.4 Mensajes transmitidos por el bus

Antes de comenzar con la descripción de los protocolos de comunicación a través del bus y para facilitarla, conviene conocer los distintos mensajes que se pueden transmitir por el bus, combinando las señales que lo componen.

A continuación se explica brevemente el significado de cada uno de los mensajes transmitidos por el bus. La combinación de señales que configuran el mensaje se detallan en la tabla 3.7, que al mismo tiempo sirve de resumen de la lista de mensajes.

Se pueden distinguir dos tipos de mensajes, según

requieran una o más líneas para ser transmitidas por el bus. En la tabla se indican con una U los que usan una sola línea y con una M los que usan más de una.

3.5.4.1 *Atención: ATN*

Este mensaje consiste en activar la señal de atención del bus, operación ésta que sólo puede ser efectuada por el dispositivo controlador del bus. Cuando el controlador está enviando este mensaje, todos los demás dispositivos deben permanecer en silencio, es decir, que no pueden enviar ningún mensaje por el bus, salvo para indicar al controlador que se ha recibido un mensaje.

El mensaje de atención indica a todos los dispositivos conectados al bus que el controlador está enviando por el bus de datos un mensaje que puede ser de interés para cualquier dispositivo.

3.5.4.2 *Mandatos universales: UCG (Universal Command Group)*

Son los mandatos que deben ser ejecutados por todos los dispositivos capaces de interpretarlos. Se entiende por dispositivo capaz de interpretar un mensaje, aquel que tiene implantada la función que le capacita para ello, sin ningún otro requisito.

Los mandatos universales son ocho:

3.5.4.2.1 Limpiar dispositivo: DCL

Activa la función de borrar el dispositivo (DC).

3.5.4.2.2 Bloqueo local: LLO (Local Lockout)

Al recibir la función RL este mensaje, impide que el dispositivo pueda cambiar de control remoto a control local, como consecuencia de una orden local.

3.5.4.2.3 Desmontar la encuesta paralelo: PPU (Parallel Poll Unconfigure)

Retira el permiso para identificarse como peticionario del servicio del bus, cuando el controlador hace una encuesta en paralelo, a todos los dispositivos que lo tenían en ese momento.

3.5.4.2.4 Inicio de encuesta serie: SPE (Serial Poll Enable)

Indica a todos los dispositivos que les va a preguntar su estado por turno, uno tras otro, para determinar si han solicitado el servicio del bus.

3.5.4.2.5 Fin de encuesta serie: SPD (Serial Poll Disable)

Indica a todos los dispositivos que ha terminado de preguntar los estados internos.

3.5.4.2.6 Identificación: IDY (Identify)

Este mensaje lo transmite el controlador activando las señales EOI y ATN al unísono. Indica la realización de una encuesta paralelo. Deben responder todos los dispositivos a los que se ha validado la facultad de hacerlo, indicando si solicitan o no el servicio del bus.

3.5.4.2.7 Limpiar la interconexión: IFC (Interface Clear)

Este mensaje inicializa todas las funciones de las interfaces conectadas al bus.

3.5.4.2.8 Validar control remoto: REN (Remote Enable)

Este mensaje se debe enviar junto a la dirección de oyente del dispositivo al que se quiere dar la orden de actuar en función de los mensajes recibidos a través del bus.

También se puede enviar este mensaje junto al de bloqueo local, lo que provoca el paso de todos los dispositivos a un estado en que sólo el controlador puede ordenarles el cambio de control remoto a local, o viceversa.

3.5.4.3 *Mandatos selectivos: ACG (Addressed Command Group)*

Son los mensajes enviados por el controlador que sólo pueden ser interpretados por él o los dispositivos que han sido previamente seleccionados para hacerlo. La forma en que esta selección se realiza se indicará más adelante.

Los mandatos direccionados son los siguientes:

3.5.4.3.1 Disparo de un grupo: GET (Group Execute Trigger)

Pone en marcha las operaciones básicas de los dispositivos previamente direccionados como oyentes.

3.5.4.3.2 Pasar a local: GTL (Go to Local)

Deja bajo control local a los dispositivos previamente direccionados como oyentes.

3.5.4.3.3 Montar la encuesta paralelo: PPC (Parallel Poll Configure)

Indica, al o a los dispositivos que están seleccionados como oyentes, que se les va a autorizar o desautorizar para responder cuando se haga una encuesta paralelo.

3.5.4.3.4 Limpiar los dispositivos seleccionados: SDC (Selected Device Clear)

Inicializa (DC) los dispositivos previamente seleccionados como oyentes.

3.5.4.3.5 Toma el control: TCT (Take control)

El controlador envía este mensaje para transferir el control del bus a otro dispositivo, que habrá sido seleccionado previamente como locutor.

3.5.4.4 *Direcciones: AD*

Para seleccionar un dispositivo como locutor o como oyente, ya sea para inicializar un diálogo, ya para transmitirle a continuación un mandato selectivo, se emplean los mensajes de dirección.

Todos los dispositivos que tienen la función que les permite actuar de locutores tienen una dirección de locutor distinta. Todos los que tienen la función de oyente tienen su propia dirección de oyente. Los dispositivos, que tienen las dos funciones tienen la misma dirección en ambas.

El controlador puede enviar cuatro mensajes de dirección distinta:

3.5.4.4.1 Dirección de oyente: MLA (My Listen Address)

Cada uno de los dispositivos conectados al bus detectan, por los bits 6 y 7 de los datos, que la dirección es de oyente.

La dirección propiamente dicha la constituyen los bits 1 a 5 del bus de datos.

Puede haber más de un dispositivo con la misma dirección de oyente.

Nemotécnico	Nombre del mensaje		TIPO	CLASE	DIO 8	7	6	5	4	3	2	DIO 1	DAV NRFD NDAC	ATN	EOI	SRQ	IFC	REN
					Líneas del bus y códigos que corresponden al mensaje.													
ACG	Grupo de órdenes selectivas	(Nota 10)	M	AC	X	0	0	0	X	X	X	X	XXX	X	X	X	X	X
ATN	Atención		U	UC	X	X	X	X	X	X	X	X	XXX	1	X	X	X	X
DAB	Octeto de datos	(Notas 1,9)	M	DD	D8	D7	D6	D5	D4	D3	D2	D1	XXX	X	X	X	X	X
DAC	Dato aceptado		U	HS	X	X	X	X	X	X	X	X	XX0	X	X	X	X	X
DAV	Dato válido		U	HS	X	X	X	X	X	X	X	X	1XX	X	X	X	X	X
DCL	Limpiar el dispositivo	(Nota 10)	M	UC	X	0	0	1	0	1	0	0	XXX	X	X	X	X	X
END	Fin	(Notas 9,11)	U	ST	X	X	X	X	X	X	X	X	XXX	X	1	X	X	X
EOS	Fin de la cadena	(Notas 2,9)	M	DD	E8	E7	E6	E5	E4	E3	E2	E1	XXX	X	X	X	X	X
GET	Disparo de un grupo	(Nota 10)	M	AC	X	0	0	0	1	0	0	0	XXX	X	X	X	X	X
GTL	Pasar a local	(Nota 10)	M	AC	X	0	0	0	0	0	0	1	XXX	X	X	X	X	X
IDY	Identificación	(Notas 10, 11)	U	UC	X	X	X	X	X	X	X	X	XXX	X	1	X	X	X
IFC	Limpiar la interconexión		U	UC	X	X	X	X	X	X	X	X	XXX	X	X	X	1	X
LAG	Grupo de direcciones de oyente	(Nota 10)	M	AD	X	0	1	X	X	X	X		XXX	X	X	X	X	X
LLO	Bloqueo Local	(Nota 10)	M	UC	X	0	0	1	0	0	0	1	XXX	X	X	X	X	X
MLA	Mi dirección de oyente	(Notas 3, 10)	M	AD	X	0	1	L5	L4	L3	L2	L1	XXX	X	X	X	X	X
MTA	Mi dirección de locutor	(Notas 4, 10)	M	AD	X	1	0	T5	T4	T3	T2	T1	XXX	X	X	X	X	X
MSA	Mi dirección secundaria	(Notas 5, 10)	M	SE	X	1	1	S5	S4	S3	S2	S1	XXX	X	X	X	X	X
NUL	Octeto nulo		M	DD	0	0	0	0	0	0	0	0	XXX	X	X	X	X	X
OSA	Otra dirección secundaria	(Nota 10)	M	SE	(OSA = SCG ∧ $\overline{\text{MSA}}$)													
OTA	Otra dirección de locutor	(Nota 10)	M	AD	(OTA = TAG ∧ $\overline{\text{MTA}}$)													
PCG	Grupo de órd. primarias	(Nota 10)	M	—	(PCG = ACG V UCG V LAG V TAG)													
PPC	Montar encuesta en paralelo	(Nota 10)	M	AC	X	0	0	0	0	1	0	1	XXX	X	X	X	X	X
PPE	Validar encuesta en paralelo	(Notas 6, 10)	M	SE	X	1	1	0	S	P3	P2	P1	XXX	X	X	X	X	X
PPD	Invalidar encuesta en paralelo	(Notas 7, 10)	M	SE	X	1	1	1	D4	D3	D2	D1	XXX	X	X	X	X	X
PPR1	Respuesta a encuesta en paralelo 1		U	ST	X	X	X	X	X	X	X	1	XXX	X	X	X	X	X
PPR2	Respuesta a encuesta en paralelo 2		U	ST	X	X	X	X	X	X	1	X	XXX	X	X	X	X	X
PPR3	Respuesta a encuesta en paralelo 3		U	ST	X	X	X	X	X	1	X	X	XXX	X	X	X	X	X
PPR4	Respuesta a encuesta en paralelo 4		U	ST	X	X	X	X	1	X	X	X	XXX	X	X	X	X	X
PPR5	Respuesta a encuesta en paralelo 5		U	ST	X	X	X	1	X	X	X	X	XXX	X	X	X	X	X
PPR6	Respuesta a encuesta en paralelo 6		U	ST	X	X	1	X	X	X	X	X	XXX	X	X	X	X	X
PPR7	Respuesta a encuesta en paralelo 7		U	ST	X	1	X	X	X	X	X	X	XXX	X	X	X	X	X
PPR8	Respuesta a encuesta en paralelo 8		U	ST	1	X	X	X	X	X	X	X	XXX	X	X	X	X	X
PPU	Desmontar encuesta en paralelo	(Nota 10)	M	UC	X	0	0	1	0	1	0	1	XXX	X	X	X	X	X
REN	Validar control remoto		U	UC	X	X	X	X	X	X	X	X	XXX	X	X	X	X	1
RFD	Listo para dato		U	HS	X	X	X	X	X	X	X	X	X0X	X	X	X	X	X
RQS	Servicio pedido	(Nota 9)	U	ST	X	1	X	X	X	X	X	X	XXX	X	X	X	X	X
SCG	Grupo de órdenes secund	(Nota 10)	M	SE	X	1	1	X	X	X	X	X	XXX	X	X	X	X	X
SDC	Limpiar los dispositivos seleccionados	(Nota 10)	M	AC	X	0	0	0	0	1	0	0	XXX	X	X	X	X	X
SPD	Fin encuesta serie	(Nota 10)	M	UC	X	0	0	1	1	0	0	1	XXX	X	X	X	X	X
SPE	Inicio encuesta serie	(Nota 10)	M	UC	X	0	0	1	1	0	0	0	XXX	X	X	X	X	X
SRQ	Petición de servicio		U	ST	X	X	X	X	X	X	X	X	XXX	X	X	1	X	X
STB	Octeto de estado.	(Notas 8, 9)	M	ST	S8	X	S6	S5	S4	S3	S2	S1	XXX	X	X	X	X	X
TCT	Toma el control	(Nota 10)	M	AC	X	0	0	0	1	0	0	1	XXX	X	X	X	X	X
TAG	Grupo de direcc. de locutor	(Nota 10)	M	AD	X	1	0	X	X	X	X	X	XXX	X	X	X	X	X
UCG	Grupo de órd. universales	(Nota 10)	M	UC	X	0	0	1	X	X	X	X	XXX	X	X	X	X	X
UNL	No oir	(Nota 10)	M	AD	X	0	1	1	1	1	1	1	XXX	X	X	X	X	X
UNT	No hablar	(Nota 10)	M	AD	X	1	0	1	1	1	1	1	XXX	X	X	X	X	X

Tabla 3.7 Codificación de los mensajes remotos. SIMBOLOGÍA: U = Mensaje de una sola línea. M = Mensaje multilínea. AC = Mandatos selectivos (Addressed command). AD = Direcciones de locutor u oyente. DD = Dependiente del dispositivo. HS = Manipulación de datos (Handsake). UC = Mandato universal. SE = Mensaje secundario. ST = Mensaje de estado. (Ver notas página siguiente).

3.5.4.4.2 No oír: UNL (Unlisten)

La dirección de oyente más alta de los posibles, 11111, no corresponde a ningún dispositivo, y es interpretada por todos los dispositivos, que estaban seleccionados como oyentes, como una orden para que dejen de atender a los mensajes transmitidos por el bus.

3.5.4.4.3 Dirección de locutor: MTA u OTA

Cuando el controlador envía por el bus una dirección de locutor, el dispositivo al que corresponde esa dirección queda seleccionado como locutor para futuras comunicaciones a través del bus. El dispositivo que estaba actuando de locutor detecta que se ha transmitido otra dirección de locutor y deja de actuar como tal.

Un mismo mensaje de dirección de locutor es interpretado de dos formas distintas por el dispositivo que actuaba de locutor y por el propietario de la dirección transmitida. En la tabla 3.7 se han reflejado estas dos interpretaciones como dos mensajes distintos, con las siglas:

MTA: Mi dirección de locutor (My Talk Address).

OTA: Otra dirección de locutor (Other Talk Address).

Una dirección de locutor solo corresponde a un dispositivo.

3.5.4.4.4 No hablar: UNT (Untalk)

La dirección 31 es interpretada por los dispositivos capaces de hablar (actuar de locutor), como una orden para dejar de transmitir mensajes.

3.5.4.5 *Mensajes secundarios: SR*

Son los mensajes que complementan a otro, enviado previamente. Son los siguientes:

3.5.4.5.1 Dirección secundaria: MSA u OSA

Al hablar de las funciones T y L (locutor y oyente), ya se apuntaba que la dirección podría tener 5 ó 10 bits y ser transmitida en uno o dos octetos. La dirección secundaria constituye el segundo octeto de una dirección de locutor o de oyente.

El controlador puede enviar una dirección de oyente primaria y a continuación una serie de direcciones secundarias, seleccionando de esta forma a varios dispositivos con la función LE. Un dispositivo considera su dirección secundaria (MSA = My Secondary Address), si el último mandato primario que ha recibido fue su dirección primaria de locutor o de oyente. En caso contrario la considera otra dirección secundaria (OSA = Other Secondary Address).

3.5.4.5.2 Validación de encuesta paralelo: PPE (Parallel Poll Enable)

El mensaje universal de asignación de encuesta paralelo (PPC) va seguido de un permiso para responder a las encuestas paralelo, para el último dispositivo que ha sido direccionado como oyente.

A este dispositivo se le asigna uno de los ocho bits de datos, para que indique si requiere o no el servicio del bus cuando se haga la encuesta paralelo. Esta asignación se hace mediante un código de tres bits. (Pi). El código de este mensaje incluye además un bit, *S*, de sentido. Cuando el dispositivo direccionado reciba el mandato de encuesta paralelo, comparará su bit de estado interno con el bit *S* recibido en el último PPE, y sólo si coinciden indicará que requiere el uso del bus.

3.5.4.5.3 Invalidar encuesta paralelo: PPD (Parallel Poll Diable)

Para poder asignar un bit del bus de datos a un dispositivo es necesario haberlo desasignado al que lo usaba hasta el momento. Para ello se pueden usar dos mandatos, el PPU, que desasigna todos los bits al mismo tiempo, y el PPD, que desasigna sólo los bits de los dispositivos actualmente direccionados como oyentes.

Lo mismo que en el caso anterior este mensaje debe ser emitido después del mensaje PPC para que tenga sentido.

3.5.4.6 *Mensajes de estado: ST*

Son los mensajes enviados por los dispositivos para indicar su estado interno.

3.5.4.6.1 Fin: END

Indica el final de una transmisión de datos entre un locutor y uno o varios dispositivos oyentes.

3.5.4.6.2 Octeto de estado: STB o RQS

Cuando el controlador hace una encuesta en serie, al interrogar a un dispositivo éste responde enviando por el bus de datos una palabra de estado. En general este mensaje recibe el nombre de octeto de estado, y en la tabla está simbolizado con las siglas STB (Status byte). En aquélla se considera además un caso particular de este mensaje, servicio pedido, que se indica con RQS (Request Service). Este mensaje, RQS, es el mismo mensaje de estado con el bit 7 a 1, lo que indica al controlador que el dispositivo ha pedido el servicio del bus.

3.5.4.6.3 Petición de servicio: SRQ (Service request)

Se envía forzando a una línea especial, SRQ, en paralelo con el mensaje que se esté enviando en ese instante. En el apartado 3.5.6 se desarrolla el conjunto de operaciones desencadenadas por esta petición de servicio.

3.5.4.6.4 Respuesta a encuesta paralelo: PPRi

Cuando el controlador envía el mensaje IDY, indicando la realización de una encuesta paralelo, cada uno de los ocho

NOTAS: 1. D1 a D8 especifican los bits de datos dependientes del dispositivo. 2. E1 a E8 especifican el código dependiente del dispositivo, empleado para indicar el mensaje EOS. 3. L1 a L5 especifican los bits dependientes del dispositivo correspondientes a la dirección del oyente. 4. T1 a T5 especifican los bits dependientes del dispositivo correspondientes a la dirección de locutor. 5. S1 a S5 especifican los bits dependientes del dispositivo correspondiente a la dirección secundaria. 6. Se indica el valor que debe tomar el bit de estado, en la respuesta a PPR, para que sea considerado activo. P1 a P3 especifican el mensaje PPR que se debe enviar cuando se ejecute la encuesta en paralelo (ej., P3 = 1; P2 = 1, P1 = 0 indican el mensaje PPR6). 7. D1 a D4 especifican bits indiferentes que ceben ser enviados todos a cero, pero que no tienten por qué ser decodificados por el dispositivo receptor. 8. S1 a S6, S8 especifican el estado del dispositivo. (DIO7 se usa para el mensaje RQS). 9. El mensaje debe ser ignorado si el dispositivo no ha sido seleccionado como oyente. 10. El mensaje debe ser ignorado si no se recibe simultáneamente el mensaje ATN. 11. El mensaje IDY se envía cuando coincide con el ATN. Se envía el mensaje END cuando no coincide simultáneamente en el bus con el mensaje ATN.

ESTADOS LINEAS: 0 = Cero lógico. Nivel alto. 1 = Uno lógico. Nivel bajo. X = Indiferente, cuando corresponde a un mensaje recibido. No se debe controlar, cuando corresponde a un mensaje emitido.

Tabla 3.7 (Continuación). Notas

dispositivos asignados responden con uno de estos mensajes, colocando el bit correspondiente a 1 si requieren el uso del bus. Dado que cada dispositivo sólo actúa sobre uno de los ocho bits del bus de datos, los ocho mensajes se pueden enviar al mismo tiempo, razón por la cual se dice que la encuesta es en paralelo.

3.5.4.7 *Mensajes de manipulación de datos: HS (Handshake)*

Estos mensajes son los empleados por las funciones AH y SH para sincronizarse cada vez que la segunda envía un octeto a la primera. Más adelante (§ 3.5.5) veremos la interrelación entre estos mensajes, que son tres:

3.5.4.7.1 Dato aceptado: DAC (Data accepted)

La función AH del dispositivo oyente indica que ya ha almacenado el octeto que se le está enviando por el bus.

3.5.4.7.2 Dato válido: DAV (Data valid)

La función SH del dispositivo locutor indica que los bits enviados por el bus de datos ya están estables y que pueden ser almacenados por los oyentes.

3.5.4.7.3 Listo para dato: RFD (Ready for data)

La función AH del oyente indica que ya ha tratado el último dato almacenado y que puede tomar un nuevo octeto del bus de datos.

3.5.4.8 *Mensajes dependientes del dispositivo: DD (Device Dependent)*

Son los mensajes de datos (DAB = Data byte) que se intercambian locutor y oyente, una vez el controlador les ha transferido el servicio del bus.

Se distinguen dos casos particulares:

— Fin de una cadena de datos: EOS (End of String), que es el código interpretado por los dos dispositivos como fin del diálogo.

— Nulo: NUL, que es un octeto con todo ceros.

En la norma del IEEE se recomienda que estos datos estén codificados con el código ISO-7 (tabla 3.8), transmitiendo el bit 7 por DIO7 y el bit 1 por DIO1. Se puede usar también el subconjunto de 6 bits. Si se usa un código binario, se debe enviar el bit de mayor peso por DIO8.

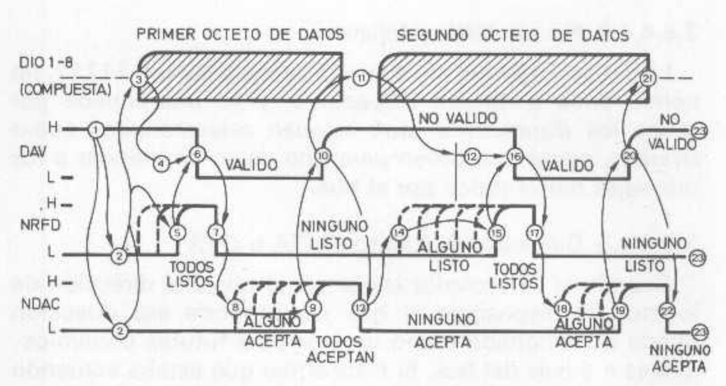
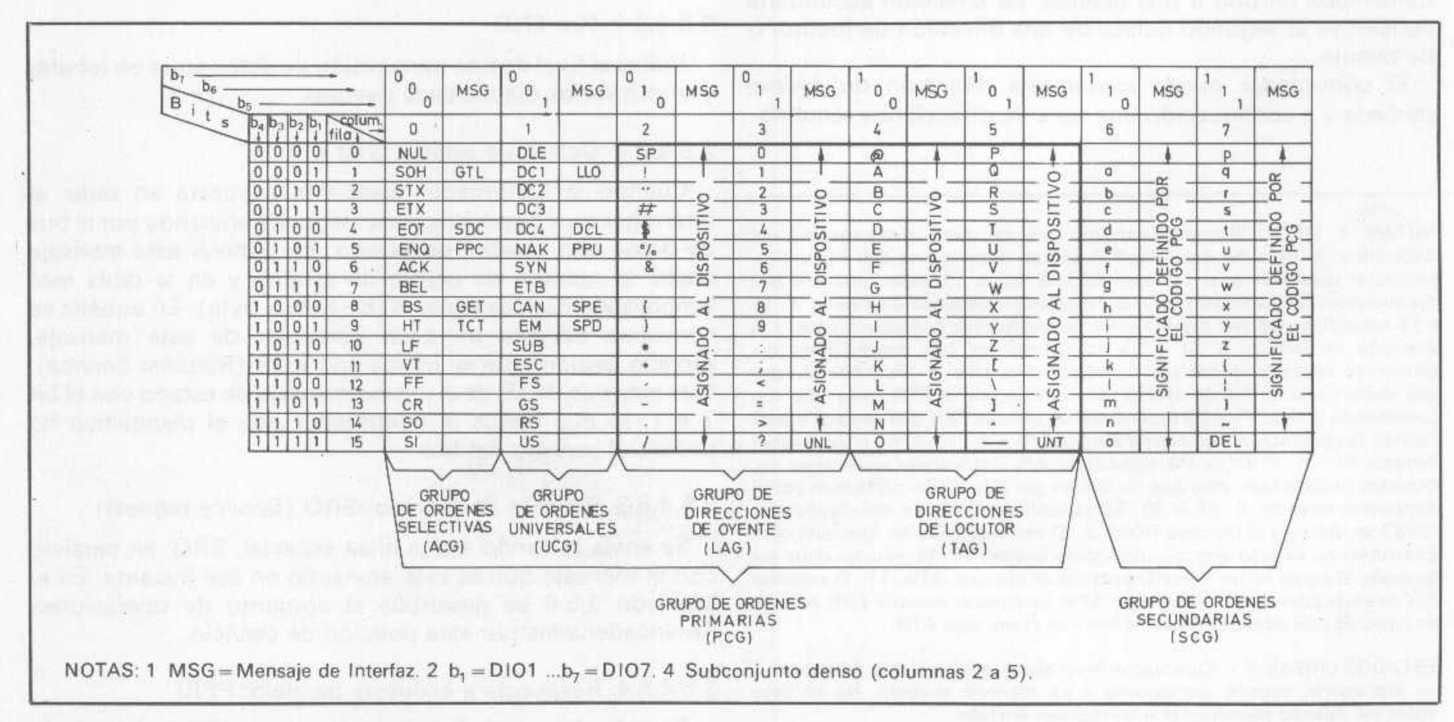


Figura 3.19 Cronograma de las señales empleadas en el protocolo de manipulación de datos cuando hay un locutor y varios oyentes.

3.5.5 Transferencia de datos

La transferencia de datos se realiza a través de las líneas DIOi. La sincronización entre emisor y receptor se consigue con las señales DAV, NRFD y NDAC, tal como ya se ha indicado. El protocolo de la transferencia se realiza de acuerdo con el cronograma de la figura 3.19.

Cuando el locutor ha situado el dato en el bus, lo indica enviando el mensaje DAV. (Dato válido). Los oyentes, al detectarlo, dejan de enviar el mensaje RFD (listo para dato) y, tras tomarse el tiempo necesario, van indicando que han aceptado el dato, enviando el mensaje DAC (Dato aceptado). El bus realiza una operación AND-cableada con esta



b_4	b_3	b_2	b_1	fila \ colum.	0 ($b_7b_6b_5$ = 000)	MSG	1 (001)	MSG	2 (010)	MSG	3 (011)	MSG	4 (100)	MSG	5 (101)	MSG	6 (110)	MSG	7 (111)	MSG
0	0	0	0	0	NUL		DLE		SP	ASIGNADO AL DISPOSITIVO	0	ASIGNADO AL DISPOSITIVO	@	ASIGNADO AL DISPOSITIVO	P	ASIGNADO AL DISPOSITIVO		SIGNIFICADO DEFINIDO POR EL CODIGO PCG	p	SIGNIFICADO DEFINIDO POR EL CODIGO PCG
0	0	0	1	1	SOH	GTL	DC1	LLO	!		1		A		Q		a		q	
0	0	1	0	2	STX		DC2		"		2		B		R		b		r	
0	0	1	1	3	ETX		DC3		#		3		C		S		c		s	
0	1	0	0	4	EOT	SDC	DC4	DCL	$		4		D		T		d		t	
0	1	0	1	5	ENQ	PPC	NAK	PPU	%		5		E		U		e		u	
0	1	1	0	6	ACK		SYN		&		6		F		V		f		v	
0	1	1	1	7	BEL		ETB		'		7		G		W		g		w	
1	0	0	0	8	BS	GET	CAN	SPE	(		8		H		X		h		x	
1	0	0	1	9	HT	TCT	EM	SPD	)		9		I		Y		i		y	
1	0	1	0	10	LF		SUB		*		:		J		Z		j		z	
1	0	1	1	11	VT		ESC		+		;		K		[		k		{	
1	1	0	0	12	FF		FS		,		<		L		\		l		¦	
1	1	0	1	13	CR		GS		-		=		M		]		m		}	
1	1	1	0	14	SO		RS		.		>		N		^		n		~	
1	1	1	1	15	SI		US		/		?	UNL	O		—	UNT	o		DEL	

GRUPO DE ORDENES SELECTIVAS (ACG) — GRUPO DE ORDENES UNIVERSALES (UCG) — GRUPO DE DIRECCIONES DE OYENTE (LAG) — GRUPO DE DIRECCIONES DE LOCUTOR (TAG)

GRUPO DE ORDENES PRIMARIAS (PCG) — GRUPO DE ORDENES SECUNDARIAS (SCG)

NOTAS: 1 MSG = Mensaje de Interfaz. 2 b_1 = DIO1 ...b_7 = DIO7. 4 Subconjunto denso (columnas 2 a 5).

Tabla 3.8. Mensajes de interfaz multilínea y código ISO-7, recomendado para las conversaciones entre locutor y oyente. Los mensajes deben estar validados por la señal ATN = 1.

señal, lo que no permite al locutor recibir el mensaje hasta que todos los oyentes lo están enviando.

Cuando el locutor detecta el mensaje DAC enviando por todos los oyentes, retira del bus el mensaje DAV.

Al detectarlo los oyentes, retiran el mensaje DAC y vuelven a enviar el mensaje RFD, cuando estén listos para almacenar un nuevo dato.

Para que el locutor detecte el mensaje RFD deben enviarlo todos los oyentes, pues el bus hace una operación AND cableada con esta señal. Cuando el locutor recibe el mensaje RFD, se inicia nuevamente el proceso.

3.5.6 Petición de servicio del bus (interrupción)

Cuando un dispositivo pone activa (1 = baja) la línea SRQ (envía el mensaje SRQ), el controlador interpreta que algún dispositivo está deseando utilizar el bus.

Como puede haber más de un dispositivo solicitando el bus en un instante determinado, el bus debe hacer una operación OR con todos los mensajes (señales) SRQ, de forma que cuando una o varias de ellas estén activas (bajas) el controlador recibe una señal activa (baja).

Ante la petición de servicio, el controlador puede responder de dos formas, para buscar el dispositivo o dispositivos peticionarios del bus: la búsqueda en paralelo y la búsqueda en serie.

3.5.6.1 *Búsqueda en paralelo*

Es la más rápida, pues permite a ocho dispositivos preseleccionados responder a la pregunta del controlador sobre si desean o no el uso del bus.

En el desarrollo de la búsqueda en paralelo se produce la siguiente secuencia de mensajes:

IDY —El controlador pregunta a los dispositivos seleccionados su estado.
PPRi —Los ocho dispositivos seleccionados ($i = 1$ a 8) responden a la pregunta al mismo tiempo, controlando cada uno de ellos un bit del bus de datos. El controlador interpreta los mensajes y da servicio al más prioritario, para lo cual debe inicializar a los interlocutores del diálogo del bus, enviando los siguientes mensajes:
UNL —Inhibe a los actuales oyentes.
LAD (1) —Direcciona el primer oyente.
⋮ ⋮
LAD (n) —Direcciona el enésimo oyente.
TAD (m) —Direcciona al locutor.
$\overline{\text{ATN}}$ —Todos los mensajes anteriores han sido validados por el mensaje ATN, tras ellos el controlador deja de enviar este mensaje, lo cual es interpretado por el locutor y los oyentes direccionados como un permiso para iniciar la transferencia de datos entre ellos. Esta transferencia finaliza con el mensaje END.

Para seleccionar los ocho dispositivos que pueden responder a una búsqueda paralelo, se desencadena la siguiente secuencia de mensajes:

(PPU) —Retira a todos los dispositivos la capacidad de responder a una encuesta en paralelo.
Desde p = 0 *hasta* 7 *hacer*
UNL —Inhibe a los actuales oyentes.
MLAi —Dirección de uno de los dispositivos a los que se va a enviar el mensaje de ...
PPC —configuración de la encuesta paralelo ...
PPEp —autorizando a responder mediante el bit p del mensaje PPRp.
Fin

Si sólo se desea cambiar uno de los dispositivos capaces de responder a la búsqueda en paralelo, se envían los siguientes mensajes:

UNL —Inhibe los actuales oyentes.
MLAi —Selecciona como oyente al dispositivo que se va a desactivar, indicándole que ...
PPC —se reconfigura la encuesta en paralelo y ...
PPD —él queda desautorizado a responder.
UNL —Le inhibe como oyente.
MLAj —Selecciona como oyente al dispositivo que se va a activar, indicándole que ...
PPC —se reconfigura la encuesta en paralelo y ...
PPEp —se le autoriza a responder mediante el bit p del mensaje PPRp.

3.5.6.2 *Búsqueda en serie*

Es más larga que la anterior, pero no requiere unos dispositivos tan complejos.

El controlador, tras haber recibido el mensaje SRQ, inicia una serie de preguntas a los dispositivos, mediante los mensajes:

UNL —Inhibe los actuales oyentes.
SPE —Indica a todos el inicio de una encuesta serie.
Repite
MTAi —Dirección del locutor i ...
STB —el cual devuelve el octeto de estado
Hasta STB = RQS
— Hasta que este octeto tenga a 1 el bit indicador de servicio pedido.
SPD —Indica a todos el fin de la encuesta serie.

Una vez encontrado el dispositivo que ha pedido el servicio, se le pasa el control de la misma forma que se hace en la búsqueda en paralelo.

3.6 NORMAS DE INTERCONEXION SERIE: RS-232-C

Las normas RS-232 fueron definidas por la EIA (Electrical Industry Association) en cooperación con la Bell System, los fabricantes de ordenadores y los fabricantes de modems (moduladores-demoduladores) privados con el objeto de normalizar los circuitos de interconexión, llamados circuitos de enlace del interfaz, entre el equipo terminal de datos (ETD) y el equipo de terminación del circuito de datos (ETCD).

En la actualidad la norma RS-232-C es la más usada en la comunicación serie entre los ordenadores y sus periféricos tales como impresoras, terminales de video, trazadores gráficos, modems, ... Sin embargo, tiene las limitaciones de separación entre el ETD y ETCD (aproximadamente 15 metros) y velocidades de transferencia de la información (hasta 20 kilobits/segundo).

Las normas RS-232 cubren los tres aspectos siguientes de la comunicación entre el ETD y el ETCD: características eléctricas de las señales, características mecánicas de los conectores y descripción funcional de las señales usadas

señal, lo que no permite al locutor recibir el mensaje hasta que todos los oyentes lo están enviando.

Cuando el locutor detecta el mensaje DAC enviando por todos los oyentes, retira del bus el mensaje DAV.

Al detectarlo los oyentes, retiran el mensaje DAC y vuelven a enviar el mensaje RFD, cuando estén listos para almacenar un nuevo dato.

Para que el locutor detecte el mensaje RFD deben enviarlo todos los oyentes, pues el bus hace una operación AND cableada con esta señal. Cuando el locutor recibe el mensaje RFD, se inicia nuevamente el proceso.

3.5.6 Petición de servicio del bus (interrupción)

Cuando un dispositivo pone activa (1 = baja) la línea SRQ (envía el mensaje SRQ), el controlador interpreta que algún dispositivo está deseando utilizar el bus.

Como puede haber más de un dispositivo solicitando el bus en un instante determinado, el bus debe hacer una operación OR con todos los mensajes (señales) SRQ, de forma que cuando una o varias de ellas estén activas (bajas) el controlador recibe una señal activa (baja).

Ante la petición de servicio, el controlador puede responder de dos formas, para buscar el dispositivo o dispositivos peticionarios del bus: la búsqueda en paralelo y la búsqueda en serie.

3.5.6.1 *Búsqueda en paralelo*

Es la más rápida, pues permite a ocho dispositivos preseleccionados responder a la pregunta del controlador sobre si desean o no el uso del bus.

En el desarrollo de la búsqueda en paralelo se produce la siguiente secuencia de mensajes:

IDY —El controlador pregunta a los dispositivos seleccionados su estado.
PPRi —Los ocho dispositivos seleccionados ($i = 1$ a 8) responden a la pregunta al mismo tiempo, controlando cada uno de ellos un bit del bus de datos. El controlador interpreta los mensajes y da servicio al más prioritario, para lo cual debe inicializar a los interlocutores del diálogo del bus, enviando los siguientes mensajes:
UNL —Inhibe a los actuales oyentes.
LAD (1) —Direcciona el primer oyente.
⋮ ⋮
LAD (n) —Direcciona el enésimo oyente.
TAD (m) —Direcciona al locutor.
$\overline{\text{ATN}}$ —Todos los mensajes anteriores han sido validados por el mensaje ATN, tras ellos el controlador deja de enviar este mensaje, lo cual es interpretado por el locutor y los oyentes direccionados como un permiso para iniciar la transferencia de datos entre ellos. Esta transferencia finaliza con el mensaje END.

Para seleccionar los ocho dispositivos que pueden responder a una búsqueda paralelo, se desencadena la siguiente secuencia de mensajes:

(PPU) —Retira a todos los dispositivos la capacidad de responder a una encuesta en paralelo.
Desde p = 0 *hasta* 7 *hacer*
UNL —Inhibe a los actuales oyentes.
MLAi —Dirección de uno de los dispositivos a los que se va a enviar el mensaje de ...
PPC —configuración de la encuesta paralelo ...
PPEp —autorizando a responder mediante el bit p del mensaje PPRp.
Fin

Si sólo se desea cambiar uno de los dispositivos capaces de responder a la búsqueda en paralelo, se envían los siguientes mensajes:

UNL —Inhibe los actuales oyentes.
MLAi —Selecciona como oyente al dispositivo que se va a desactivar, indicándole que ...
PPC —se reconfigura la encuesta en paralelo y ...
PPD —él queda desautorizado a responder.
UNL —Le inhibe como oyente.
MLAj —Selecciona como oyente al dispositivo que se va a activar, indicándole que ...
PPC —se reconfigura la encuesta en paralelo y ...
PPEp —se le autoriza a responder mediante el bit p del mensaje PPRp.

3.5.6.2 *Búsqueda en serie*

Es más larga que la anterior, pero no requiere unos dispositivos tan complejos.

El controlador, tras haber recibido el mensaje SRQ, inicia una serie de preguntas a los dispositivos, mediante los mensajes:

UNL —Inhibe los actuales oyentes.
SPE —Indica a todos el inicio de una encuesta serie.
Repite
MTAi —Dirección del locutor i ...
STB —el cual devuelve el octeto de estado
Hasta STB = RQS
— Hasta que este octeto tenga a 1 el bit indicador de servicio pedido.
SPD —Indica a todos el fin de la encuesta serie.

Una vez encontrado el dispositivo que ha pedido el servicio, se le pasa el control de la misma forma que se hace en la búsqueda en paralelo.

3.6 NORMAS DE INTERCONEXION SERIE: RS-232-C

Las normas RS-232 fueron definidas por la EIA (Electrical Industry Association) en cooperación con la Bell System, los fabricantes de ordenadores y los fabricantes de modems (moduladores-demoduladores) privados con el objeto de normalizar los circuitos de interconexión, llamados circuitos de enlace del interfaz, entre el equipo terminal de datos (ETD) y el equipo de terminación del circuito de datos (ETCD).

En la actualidad la norma RS-232-C es la más usada en la comunicación serie entre los ordenadores y sus periféricos tales como impresoras, terminales de video, trazadores gráficos, modems, ... Sin embargo, tiene las limitaciones de separación entre el ETD y ETCD (aproximadamente 15 metros) y velocidades de transferencia de la información (hasta 20 kilobits/segundo).

Las normas RS-232 cubren los tres aspectos siguientes de la comunicación entre el ETD y el ETCD: características eléctricas de las señales, características mecánicas de los conectores y descripción funcional de las señales usadas

Núm. de señal	Nemónico	Dirección	Breve descripción
1	AA	—	Señal de tierra
2	BA	Hacia ETCD	Transmisión de datos
3	BB	Hacia ETD	Recepción de datos
4	CA	Hacia ETCD	Petición de transmitir
5	CB	Hacia ETD	Preparado para transmitir
6	CC	Hacia ETD	Aparato de datos preparado
7	AB	—	Masa común de las señales
8	CF	Hacia ETD	Detector de señales de línea recibidas por el canal de datos
9	—	—	Reservada para la comprobación de los datos
10	—	—	Reservada para la comprobación de los datos
11	—	—	Sin asignación
12	SCF	Hacia ETD	Detector de señales de línea recibidas por el canal de reserva de datos
13	SCB	Hacia ETD	Preparado el canal de reserva para transmitir
14	SBA	Hacia ETCD	Transmisión de datos por el canal de reserva
15	DB	Hacia ETD	Temporización para los elementos de señal en la transmisión
16	SBB	Hacia ETD	Recepción de datos por el canal de reserva
17	DD	Hacia ETD	Temporización para los elementos de señal en la recepción
18	—	—	Sin asignación
19	SCA	Hacia ETCD	Petición para transmitir por el canal de reserva
20	CD	Hacia ETCD	Terminal de datos preparado
21	CG	Hacia ETDD	Detector de la calidad de las señales de datos
22	CE	Hacia ETD	Indicador de llamada
23	CH/CI	Ambas	Selector de velocidad binaria con origen ETD (H) u origen ETCD (CI)
24	DA	Hacia ETCD	Temporización para los elementos de señal en la transmisión
25	—	—	Sin asignación

Tabla 3.9

para realizar la comunicación. La letra C en RS-232-C indica la tercera y última revisión.

A lo largo de este apartado daremos solamente una descripción funcional de las señales, remitiendo al lector interesado en las otras características de la norma a la publicación original EIA1.

3.6.1 Descripción funcional de las señales

Dentro del conjunto de las señales podemos distinguir cuatro grandes grupos: de datos, de control, de temporización y las masas. En la tabla 3.9 representamos el número de la señal dentro del conector, el nemónico, el sentido de conexión entre el ETD y el ETCD y una breve descripción de sus significados.

Como norma general diremos que las señales de datos se consideran como marca (lógica 1) cuando en ellas hay una tensión negativa, y como espacio (lógica 0) cuando hay una tensión positiva. En cuanto a las señales de control y temporización se consideran en estado abierto cuando están a tensión positiva y un estado cerrado cuando están a tensión negativa.

Vamos a pasar ahora a la descripción de las señales por grupos.

3.6.1.1 *Señales de datos*

— BA. Transmisión de datos.

Es la señal usada para la transmisión de los datos entre el ETD y el ETCD. Las restricciones que debe cumplir esta señal son las siguientes:

a) El ETD deberá poner esta señal al nivel de la marca entre la transmisión de caracteres o palabras y también cuando no se transmitan datos.

b) Para que el ETD transmita datos se deberá cumplir que las señales *CB*, *CC*, *CD* y *CA* estén en el estado abierto. Normalmente, en los conectores comerciales se reconoce esta señal como TXD (datos a transmitir).

—BB. Recepción de datos.

Es la señal usada para la transmisión de los datos entre el ETCD y el ETD. Esta señal deberá estar en la condición de marca mientras la señal *CF* esté en el estado de cerrado. En un sistema half-duplex deberá estar en la condición de marca cuando la señal *CA* esté en el estado de abierto. En los conectores comerciales se reconoce esta señal como RXD (datos a recibir).

—SBA. Transmisión de datos para el canal de reserva.

Es equivalente a *BA* pero para el canal de reserva. Dicho canal trabaja a velocidades inferiores.

—SBB. Recepción de datos para el canal de reserva.

Es equivalente a *BB* pero para el canal de reserva.

3.6.1.2 *Señales de control*

—CA. Petición de transmitir.

Esta señal es enviada desde el ETD hacia el ETCD para indicarle, cuando la pone en estado de abierto, que quiere realizar una transmisión. En un sistema de half-duplex, el estado de abierto inhibe la recepción. Cuando se realiza sobre esta señal el cambio de cerrado a abierto, el ETCD responde cambiando la señal *CB* a estado abierto. Los datos a transmitir pueden ser enviados solamente después de que el ETD detecte este cambio a estado abierto de *CB*. Si la señal *CA* es cambiada a estado de cerrado no podrá ser cambiada de nuevo a estado de abierto hasta que el ETCD responda cambiando la señal *CB* a estado de cerrado. La señal *CA* es conocida normalmente como RTS.

— CB. Preparado para transmitir.

Es enviada desde el ETCD hacia el ETD. El estado de esta señal indica si el ETCD está o no preparado para transmitir datos por el canal de datos.

El estado cerrado indica que el ETCD está en condiciones de transmitir datos por el canal de datos y el estado abierto indica lo contrario. Normalmente se conoce a esta señal como CTS.

— CC. Aparato de datos preparado.

Esta señal es enviada por el ETCD hacia el ETD. El estado de esta señal indica si el ETCD está o no preparado para funcionar. El estado es abierto sólo si el ETCD ha intentado establecer una comunicación por el canal después de haber cumplido con todas las temporizaciones necesarias y generado los tonos de respuesta. El estado de abierto no indica que exista un canal de comunicación entre el ETCD y otro ETCD remoto, sino sólo el estado del ETCD local. Normalmente se conoce como DSR.

— CD. Terminal de datos preparado.

Esta señal es enviada desde el ETD hacia el ETCD. El estado abierto en esta señal es necesario para mantener la comunicación entre el ETCD local y el ETCD remoto. Su puesta a estado cerrado indica al ETCD que deberá suspender la comunicación con el ETCD remoto al final de la transmisión que se está ejecutando en ese momento. Abreviadamente se le conoce como DTR.

— CE. Indicador de llamada.

Esta señal es enviada desde el ETCD hacia el ETD. El

estado de esta señal indica si el ETCD está o no recibiendo una llamada. El estado de abierto indica que el ETCD está recibiendo una llamada. La señal se pone a estado cerrado en el intervalo entre llamadas. Para que esta señal se ponga a estado de abierto, la señal *CD* (DTR) deberá estar en estado de abierto. Abreviadamente se le denomina RI.

— CF. Detector de señales de línea recibidas por el canal de datos.

Es enviada desde el ETCD hacia el ETD. El estado de esta señal indica si las señales de línea recibidas por el canal de datos están o no dentro de los límites especificados en la recomendación pertinente para el ETCD. El estado abierto indica que la señal recibida cumple las especificaciones requeridas. Normalmente se denomina a esta señal con DCD (detección de portadora de datos).

— CG. Detector de la calidad en la señal de datos.

Esta señal va desde el ETCD hacia el ETD. El estado de esta señal indica si existe o no cierta probabilidad de error en los datos recibidos por el canal de datos. La calidad de señal indicada se ajusta a la recomendación pertinente sobre el ETCD. El estado cerrado indica que no hay motivos para creer que se ha producido un error. El estado abierto indica que existe cierta probabilidad de error.

— CH. Selector de velocidad binaria.

Esta señal va desde el ETD hacia el ETCD. El estado de esta señal sirve para seleccionar una de las dos velocidades binarias de un ETCD síncrono o una de las dos gamas de velocidades binarias en un ETCD asíncrono.

El estado cerrado causa la selección de la velocidad binaria o de la gama de velocidades binarias más elevada. El estado abierto selecciona la más baja.

— CI. Selector de velocidad binaria.

Esta señal va desde el ETCD hacia el ETD. El estado de esta señal sirve para la selección de la velocidad binaria o de la gama de velocidades binarias en el ETD en función de la velocidad binaria utilizada en un ETCD síncrono con dos velocidades binarias o de la gama de velocidades binarias utilizadas en un ETCD asíncrono con dos gamas de velocidades binarias. El estado cerrado causa la selección de la velocidad binaria o gama de velocidades binarias más alta. El estado abierto selecciona la más baja.

En una conexión dada, sólo existirá una de las dos señales anteriores.

— SCA. Petición para transmitir por el canal de reserva.

Esta señal va desde ETD hacia ETCD. Su función es equivalente a la de la señal *CA* (petición de transmitir) pero para el canal de reserva.

— SCB. Preparado el canal de reserva para transmitir.

Esta señal va desde el ETCD hacia el ETD. Su función es equivalente a la de la señal *CB* pero para el canal de reserva.

— SCF. Detector de señales de línea recibidas por el canal de reserva de datos.

Esta señal va desde el ETCD hacia el ETD. Su función es la misma que la señal *CF* pero para el canal de reserva.

3.6.1.3 *Señales de temporización*

— DA. Temporización, para los elementos de señal, en la transmisión.

Es una señal que va desde el ETD hacia el ETCD. El cambio de estado de abierto a cerrado en esta señal le indica al ETCD el centro de cada bit a transmitir.

— DB. Temporización, para los elementos de señal, en la transmisión.

Es una señal que va desde el ETCD hacia el ETD. El ETD deberá cambiar el estado de la línea BA (datos a transmitir) cuando se produzca una transición de estado cerrado a abierto en esta señal *DB*.

—DD. Temporización para los elementos de señal en la recepción.

Esta señal va desde el ETCD hacia el ETD. La transición del estado abierto a cerrado en esta línea indica al ETD el centro del bit en la línea BB (datos recibidos). Esta señal será usada en el ETD para muestrear los datos recibidos.

3.6.1.4 *Señales de masa*

— AA. Señal de tierra.

Por esta señal se conectan las masas generales del ETD y ETCD.

—AB. Masa común de las señales.

Es la señal de tierra o retorno común de forma que provee el potencial de referencia para todas las señales RS-232-C (excepto para la *AA*).

Los términos castellanos empleados en la descripción de las señales anteriores, así como algunas frases, han sido sacados de las recomendaciones V-24. Dicha recomendación es del CCITT y coincide en muchos aspectos con la RS-232-C. El lector interesado puede consultar el tomo VIII.1 (libro naranja) sobre transmisión de datos por la red telefónica CCI.

Actualmente, la norma EIA RS-232-C es tal vez la más popular para las interfaces de comunicación serie. Sin embargo, la EIA ha normalizado nuevas normas que tienden a cubrir las limitaciones de la RS-232-C en cuanto a máxima distancia y velocidad de transmisión. Estas normas se denominan RS-449, RS-422-A y RS-423-A.

La norma RS-449 especifica las características mecánicas de los conectores y las descripciones funcionales de las señales. En cuanto a las variaciones con la RS-232-C diremos que permite una velocidad máxima de 2 Mbits/segundo, incluye 10 nuevas señales anulando tres de la RS-232-C y vuelve a redefinir algunas señales. El lector interesado puede consultar las publicaciones señaladas como EIA2.

Las normas RS-422-A y RS-423-A se refieren a las características eléctricas de las señales. La idea, como vemos, ha sido separar en diferentes normas las características eléctricas de las señales y su significado (ambas, junto con las características mecánicas, estaban dentro de la RS-232-C) de forma que una norma funcional, tal como RS-449, pueda usar diferentes normas eléctricas, tales como RS-422-A y RS-423-A, dependiendo de parámetros tales como longitud del canal, velocidad de transmisión y ruido en el canal; y que diferentes normas funcionales puedan usar una misma norma de características eléctricas.

La norma RS-422-A usa señales balanceadas y la RS-423-A usa señales no balanceadas en los circuitos del interfaz. El lector interesado puede consultar ambas normas en las publicaciones EIA3 y EIA4, respectivamente.

Apéndice III.1 EL BUS S-100 (ALTAIR)

número de la línea	nombre simból. de la señal	*Descripción y características de la señal*
1	+8 V	Tensión (no regulada) en el bus, que debe ser regulada a 5 V para alimentar a los módulos del sistema.
2	+18 V	Tensión positiva pre-regulada.
3	XRDY	«External Ready»; entrada de «preparado» externa para la CPU (ver n.º 12)

número de la línea	nombre simból. de la señal	*Descripción y características de la señal*
4	VI0	Línea 0 de interrupción vectorizada.
5	VI1	Línea 1 de interrupción vectorizada.
6	VI2	Línea 2 de interrupción vectorizada.
7	VI3	Línea 3 de interrupción vectorizada.
8	VI4	Línea 4 de interrupción vectorizada.
9	VI5	Línea 5 de interrupción vectorizada.
10	VI6	Línea 6 de interrupción vectorizada.
11	VI7	Línea 7 de interrupción vectorizada.
• 12	XRDY2	«External Ready 2»; es una segunda línea de «preparado» externa, similar a la n.º 3.
13		Sin definir.
14		Sin definir.
15		Sin definir.
16		Sin definir.
17		Sin definir.
18	$\overline{\text{STAT DSB}}$	«Status Disable»; inhabilitar estado, permite a los tampones para las 8 líneas de estado su paso a tres-estados.
19	$\overline{\text{C/C DSB}}$	«Command/Control Disable»; Inhabilitar Control/Mandatos, similar a la anterior para las 6 líneas de salida de control
• 20	UNPROT	«Unprotect»; Quitar protección, es una entrada a la báscula biestable de protección de una tarjeta de memoria.
• 21	SS	«Single Step»; Paso a paso, indica que la máquina está en proceso de realizar operación paso a paso (Báscula SS a uno).
22	ADD DSB	«Address Disable»; Inhabilitar Dirección, permiso para pasar a tres estados las 16 líneas de dirección.
23	DO DSB	«Data Out Disable»; Inhabilitar Salida de Datos, idem. para las 8 líneas de datos de salida.
• 24	$\phi 2$	Fase 2 del reloj.
• 25	$\phi 1$	Fase 1 del reloj.
26	PHLDA	«Hold Acknowledge»; Aceptar bloqueo, señal de salida del procesador que responde a HOLD e indica que los buses de datos y direcciones pasarán a estado de alta impedancia y el procesador a estado bloqueo una vez completado el ciclo en curso.
• 27	PWAIT	«Wait»; Espera, aparece en respuesta al paso de Ready a nivel bajo e indica que el procesador entrará en una serie de estados de espera (de 0,5 μs) hasta que suba la señal Ready.
• 28	PINTE	«Interrupt Enable»; Habilitar Interrupción, indica que se permiten interrupciones, según lo que determine el contenido de la báscula interna de interrupciones de la CPU (que puede inhibirlas o no).
29	A5	Línea 5 de dirección.
30	A4	Línea 4 de dirección.
31	A3	Línea 3 de dirección.

número de la línea	nombre simból. de la señal	*Descripción y características de la señal*
32	A15	Línea 15 de dirección. (Bit más significativo).
33	A12	Línea 12 de dirección.
34	A9	Línea 9 de dirección.
+ 35	DO1	Línea 1 de datos de salida.
+ 36	DO0	Línea 0 de datos de salida. (Bit menos significativo).
37	A10	Línea 10 de dirección.
+ 38	DO4	Línea 4 de datos de salida.
+ 39	DO5	Línea 5 de datos de salida.
+ 40	DO6	Línea 6 de datos de salida.
+ 41	DI2	Línea 2 de datos de entrada
+ 42	DI3	Línea 3 de datos de entrada.
+ 43	DI7	Línea 7 de datos de entrada. (Bit más significativo).
44	SM1	«Machine cycle 1»; ciclo de 1 máquina, indica que el procesador se halla en ciclo de extracción del primer octeto de una instrucción.
45	SOUT	«Output»; Salida, indica que el bus de direcciones contiene la de un dispositivo de salida y que el bus de datos contendrá los datos de salida cuando se active PWR (ver n.º 77).
46	SINP	«Input»; Entrada, indica que el bus de direcciones contiene la de un dispositivo de entrada y que los datos de entrada deberán colocarse en el bus cuando se active PDBIN (n.º 78).
47	SMEMR	«Memory Read»; Leer en Memoria, indica que el bus de datos se va a usar para extraer datos de memoria.
48	SHLTA	«Halt»; Detener, indica reconocimiento de una instrucción Halt.
+49	CLOCK	«Clock»; Reloj; salida invertida de reloj ϕ Z (n.º 24).
50	GND	«Ground»; Masa del sistema.
51	+8 V	Entrada no regulada para obtener 5 V a través de reguladores.
•52	−18 V	Tensión negativa pre-regulada.
• 53	SSWI	«Sense switch input»; Entrada de conmutadores, indica que habrá una entrada de datos desde los mismos. La lógica de control/visualización la utiliza para 1.º) habilitar excitadores; 2.º) habilitar las entradas FDI0-7; 3.º) inhabilitar los excitadores de entrada de datos a CPU.
+ 54	EXT CLR	«External Clear»; Puesta a cero externa, señal para los dispositivos de E/S.
• 55	RTC	«Real-time Clock»; Reloj de tiempo real, señal de 60 Hz usada como base de tiempos
• 56	STSTB	«Status Strobe»; Validación de Estado, señal suministrada por el generador de reloj, principalmente para que se ponga el estado («cerrojo» 8212) en cuanto sea posible.
• 57	DIG1	«Data Input Gate 1»; Puerta 1 de Entrada de Datos, determina el conjunto de excitadores de entrada de datos que toma el control del bus bidireccional de datos (los de la CPU o los de control/visualización).

número de la línea	nombre simból. de la señal	*Descripción y características de la señal*
* 58	FRDY	«Front Panel Ready»; Panel frontal preparado, señal que permite al panel frontal controlar las líneas de Preparado para la CPU.
* 59		Sin definir.
* 60		Sin definir.
* 61		Sin definir.
* 62		Sin definir.
* 63		Sin definir.
* 64		Sin definir.
65		Sin definir.
66		Sin definir.
* 67		Sin definir.
68	MWRITE	«Memory Write»; Escribir en Memoria, indica que los datos presentes en el bus deben escribirse en la posición direccionada.
* 69	PS	«Protect Status»; Estado protección, indica la situación de la báscula biestable de protección de memoria.
* 70	PROT	«Protect»; Protección, señal de entrada a la báscula biestable de protección de memoria.
* 71	RUN	«Run»; Ejecutar, indica la reposición a cero de la báscula RUN, es decir que la máquina está en modo ejecución.
+ 72	PRDY	«Processor Ready»; Procesador Preparado, entrada de memoria y E/S a la circuitería de espera de la CPU.
73	PINT	«Interrupt Request»; Petición de Interrupción, el procesador reconoce una solicitud de interrupción al acabar la instrucción en curso o mientras está en parada. Si el procesador estuviese en bloqueo o la báscula de habilitación de interrupción no estuviese puesta a uno, no atenderá la petición.
74	PHOLD	«Hold»; Bloqueo, señal de entrada al procesador que solicita su paso a estado bloqueo y permite que un dispositivo externo tome el control de los buses en cuanto que el procesador complete el uso de dichos buses en el ciclo en curso.
75	PRESET	«Reset»; Reponer, al activarse se borra el contenido del contador de programa y se pone a cero el registro de instrucción.
76	PSYNC	«Sync»; Sincronizar, señal de salida del procesador que indica el comienzo de cada ciclo de máquina.
77	PWR	«Write»; Escritura, señal para escribir en memoria o para controlar una salida de un dispositivo de E/S. Mientras está activa se mantiene estable el contenido del bus de datos.

número de la línea	nombre simból. de la señal	*Descripción y características de la señal*
78	PDBIN	«Data Bus In»; señal que indica a los circuitos externos al procesador que el bus de datos está en modo entrada.
79	A0	Línea 0 de dirección. (Bit menos significativo).
80	A1	Línea 1 de dirección.
81	A2	Línea 2 de dirección.
82	A6	Línea 6 de dirección
83	A7	Línea 7 de dirección.
84	A8	Línea 8 de dirección.
85	A13	Línea 13 de dirección.
86	A14	Línea 14 de dirección.
87	A11	Línea 11 de dirección
+ 88	DO2	Línea 2 de salida de datos.
+ 89	DO3	Línea 3 de salida de datos.
+ 90	DO7	Línea 7 de salida de datos. (Bit más significativo.)
+ 91	DI4	Línea 4 de entrada de datos
+ 92	DI5	Línea 5 de entrada de datos
+ 93	DI6	Línea 6 de entrada de datos
+ 94	DI1	Línea 1 de entrada de datos
+ 95	DI0	Línea 0 de entrada de datos. (Bit menos significativo)
96	SINTA	«Interrupt Acknowledge»; Aceptar interrupción, señal de salida de estado que reconoce la presencia de una petición de interrupción.
97	SWO	«Write Out»; señal de salida de estado que indica que la operación en curso va a ser de escritura en memoria o una salida.
98	SSTACK	«Stack»; Apilar, indica que el bus de direcciones mantiene la dirección de la pila que señala el puntero.
99	POC	«Power-on Clear»; Puesta a cero al conectar la alimentación.
100	GND	«Ground»: Masa del sistema.

NOTAS:

*Indica las señales que han sido redefinidas en la nueva norma (IEEE Task 696.1.D2) que se estudia en el apartado 3.2 del texto.

+ indica las señales que han sufrido modificación conceptual en la nueva norma.

entradas-salidas en paralelo 4

Joan Navarro

4.1 GENERALIDADES

4.1.1 Tipos de entradas/salidas

El concepto de "entradas/salidas" engloba toda comunicación o intercambio de información entre la Unidad Central de Proceso y el exterior. Esta comunicación puede efectuarse por dos métodos fundamentalmente: serie y paralelo.

La entrada/salida en paralelo se basa en la transmisión simultánea de *n* bits, siendo éstos los componentes de la unidad básica de información codificada o palabra. Las distintas palabras que componen el bloque de información son transmitidas secuencialmente por las mismas vías.

La entrada/salida en serie puede ser considerada como un caso particular de la anterior en la que $n = 1$, efectuándose entonces un secuenciamiento de bits dentro de la palabra, y de palabras dentro del bloque. Dadas las problemáticas específicas de la comunicación en serie, el capítulo 5.º de este libro se dedica a su estudio en profundidad.

4.1.2 Comparación entradas/salidas serie y paralelo

La entrada/salida en paralelo transmite simultáneamente una palabra completa, tal como es utilizada en el procesador, hacia o desde unidades periféricas, de tal modo que el tiempo de transmisión de dicha palabra es mínimo con lo que la velocidad global de transmisión será únicamente penalizada por las limitaciones implícitas a la tecnología utilizada. En determinados casos esta velocidad puede ser un múltiplo o submúltiplo entero de la anterior, cuando la amplitud de la palabra transmitida sea múltiplo o submúltiplo de la amplitud de la palabra procesada expresada en el número de bits que la componen.

Como contrapartida la entrada/salida en serie exige un tiempo unitario para cada bit transmitido, por lo que la transmisión de una misma palabra requerirá los *n* tiempos elementales correspondientes, más los precisos para las informaciones de control y protección.

En cuanto a los circuitos precisos, la entrada/salida en serie exige una serialización previa y paralelización posterior de la información, que no es preciso efectuar en la entrada/salida en paralelo; como contrapartida en la comunicación paralelo se precisan las *n* vías de comunicación más las precisas para protección y control, implicando por tanto un mayor número de hilos de conexión y circuitos terminales (a todo este conjunto se le denomina "interfaz").

Como consecuencia, cuando la distancia entre conjuntos que deban comunicarse sea reducida, o bien cuando la velocidad de transmisión deba ser elevada será preferida la interfaz paralelo.

4.1.3 Sentido de transmisión

Las vías de comunicación pueden actuar unidireccional o bidireccionalmente. Líneas unidireccionales son aquellas en las que la información circula en un solo sentido, estando permanentemente definidos los elementos que actúan como transmisor y los que lo hacen como receptor, el sentido hacia o desde la unidad central depende de las características del elemento periférico.

Líneas bidireccionales son aquellas en las que la información circula secuencialmente en ambos sentidos permitiendo por tanto un diálogo entre procesador y periférico.

4.1.4 Condicionantes de entradas/salidas

En el estudio de un sistema de entradas/salidas en paralelo deben observarse una serie de aspectos que condicionan el tipo de líneas o circuitos terminales que deban utilizarse, tales como: velocidad de transmisión; distancia entre equipos; sentido de la comunicación, interferencia por ruidos externos o diafonía entre bits; número de bits componentes de la información, más el correspondiente a señales de control, más el preciso para protección de la información, que dependerá del grado de seguridad preciso y las características inherentes al periférico y a la información que éste suministra o recibe.

4.1.5 Tipos de periféricos

Los periféricos que pueden conectarse a un μP (microprocesador) pueden dividirse en dos grupos fundamentales: periféricos locales, íntimamente ligados al proceso en sí (tales como memorias RAM, ROM y FPLA, interruptores, pilotos, visualizadores numéricos y alfanuméricos, temporizadores, relojes de tiempo real, procesadores aritméticos auxiliares, etc.) y periféricos remotos que requieren circuitos específicos de control en ocasiones tanto o más complejos que la unidad central (tales como: teclados, teletipos, impresoras de alta velocidad, visualizadores en TRC, cassettes, cintas magnéticas de 1/2", discos flexibles, discos magnéticos con cabezal volante, lectores de tarjetas marcadas o perforadas, lectores y perforadores de cinta de papel, otros microprocesadores, instrumentos de medida, generadores de señales, sensores analógicos, actuadores electromecánicos, etc.).

4.2 PROTOCOLOS DE DIÁLOGO

4.2.1 Introducción

El problema fundamental radica en la enorme diferencia de velocidad entre la ejecución de las instrucciones del programa por parte del microprocesador y los ritmos de transferencia generalmente mucho más lentos impuestos por los periféricos.

Se pueden distinguir asimismo dos situaciones distintas

que se plantean en la conexión de unidades periféricas. Por una parte es preciso efectuar una transferencia de datos entre el microprocesador o su memoria y la interfaz o controlador del periférico en cuestión, efectuándose esta comunicación bajo los criterios de relación de señales y sincronismos impuestos por la propia unidad central.

La segunda situación se presenta cuando los datos deben intercambiarse entre la interfaz o controlador y el propio periférico, siendo en este caso la propia estructura de la unidad periférica quien determina la interrelación y sincronismo de señales, pudiendo adoptar muy distintas configuraciones dependiendo de las características propias relativas al periférico, en cuanto a tipo de actuación, soporte utilizado, velocidad neta e incluso del fabricante que lo ha desarrollado.

Se analizarán en primer lugar las transferencias entre la unidad central o su memoria y la interfaz o controlador.

4.2.2 Entradas/salidas por esperas

Una primera solución consiste en bloquear el proceso durante el tiempo requerido por el periférico para completar la instrucción, de tal modo que el microprocesador espera, a cada nueva instrucción de entrada/salida, que el elemento periférico esté disponible.

Según si se desea efectuar una entrada, o bien una salida, la técnica utilizada es distinta.

En el caso de una salida, lo más usual consiste en conectar los bits correspondientes a la palabra de información a los terminales de salida, de tal modo que los circuitos receptores del periférico o controlador sean excitados por el nivel lógico apropiado.

Se establece un cierto tiempo de espera para permitir que los períodos de transición asociados a la conmutación de cada bit hayan finalizado, estabilizándose los niveles lógicos, tras lo cual se emite, por una vía separada, un impulso de validación (strobe), que provoca la transferencia efectiva de los datos hacia el registro de entrada del periférico. Este tiempo de espera es imprescindible por dos razones: por una parte es muy difícil garantizar que tanto las vías de datos, como la de validación presentan tiempos de conmutación equivalentes (en cualquier caso la vía de validación ha de ser la última en conmutar, pues de lo contrario se podrían validar vías de datos de conmutación más lentas, que no hubiesen alcanzado un nivel estable, provocando transferencias erróneas de información).

Por otra parte, la rápida conmutación generada por los circuitos lógicos presenta generalmente características de oscilación amortiguada, máxime cuando la inductancia (provocada por la longitud o las características intrínsecas) de la línea es relativamente elevada; de tal modo que hasta no finalizar las posibles oscilaciones, no es recomendable validar la información, puesto que podría coincidir la señal de entrada con una cresta de oscilación (positiva o negativa) que pudiera distorsionar la correcta transferencia.

Este tiempo de espera depende tanto de las características propias de la línea, como de los circuitos terminales asociados, por lo que no puede establecerse una regla de carácter general; pero puede cifrarse en la mayoría de los casos entre 50 ns y 10 μs.

Durante el impulso de validación, evidentemente, las vías de datos deben permanecer estables.

Una vez finalizada la validación (cuya duración depende de las constantes de tiempo del circuito receptor), las vías de datos pueden, o bien desconectarse, o bien preparar para la próxima palabra que deba ser transferida.

Sin embargo, la transferencia de esta subsiguiente palabra no podrá efectuarse hasta haber finalizado la acción asociada del periférico, por lo cual los circuitos de interfaz receptores, al detectar una transferencia válida, responden activando una vía de señalización (asociada al concepto de ocupación) que, explorada por la unidad central, determina el tiempo empleado por el periférico para procesar la palabra transmitida.

Durante este período la unidad central no puede proseguir la secuencia programada, por lo que debe *esperar* (justamente a partir de esta espera proviene la definición de esta modalidad de transferencia) a que el periférico finalice las funciones asociadas.

Una vez la unidad periférica ha completado la orden recibida, avisa al microprocesador desactivando la vía de ocupación; este último detecta dicha situación y presenta la nueva palabra de información, repitiendo el proceso descrito, tantas veces cuantas palabras constituyan el bloque de información a emitir. La figura 4.1 ilustra este tipo de transferencia.

Un ejemplo puede ilustrar esta situación. Supongamos que tenemos que escribir una serie de caracteres en una impresora. Para ello se debe programar una instrucción de salida en la que los 7 bits del carácter a imprimir son enviados al periférico. Seguidamente se pone en "1" el octavo bit que se utilizará como señal de validación. La próxima instrucción pone a "0" este mismo bit, con lo que se ha generado un impulso completo. El programa siguiendo su ejecución desea transmitir un segundo carácter, pero previamente se efectúa una instrucción de bifurcación condicionada a la ocupación de la impresora, dando lugar a un bucle de espera, hasta poder emitir este segundo carácter, y de modo similar los sucesivos. Habida cuenta de la velocidad relativa de los conjuntos implicados (100.000 a 1.000.000 instrucciones por segundo del microprocesador y 10 a 100 caracteres por segundo de la impresora), aquél encontrará a ésta ocupada ya desde el segundo paso por la instrucción pocos microsegundos después, viéndose obligado a esperar muchos milisegundos hasta finalizar la impresión del primer carácter y liberarse el registro del circuito de control del periférico, para aceptar la siguiente transmisión. De este modo, el microprocesador se encuentra bloqueado durante el desarrollo completo de la entrada/salida, con la consecuencia de que el porcentaje de tiempo activo de la unidad central puede llegar a ser muy pequeño (en el ejemplo entre 0,1 % a 0,001 %).

Evidentemente el impulso de validación podía haberse generado por medios electrónicos, en el ejemplo se ha generado por programa. Esta interconexión se denomina "hand-shake" (darse la mano) que podría traducirse de modo liberal como "interactiva" y es muy simple de realización; pero como contrapartida reduce notablemente la velocidad del equipo, por lo que sólo puede emplearse en sistemas económicos, o en los que el periférico por sus

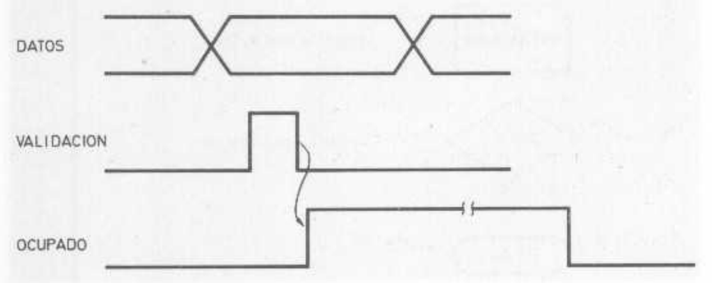


Figura 4.1 Entrada/salida por esperas

especiales características permita una elevada velocidad de transferencia o exija una dedicación continua.

En el caso de una entrada el proceso es algo distinto.

La unidad central en un punto determinado de la progresión de su programa ordena al periférico transmitir información. Este último puede tener preparada (o no tenerla) la primera palabra de datos.

Si no dispone de ella, responde activando la línea de ocupación, con lo que la unidad central detiene la ejecución de su programa y *espera* la presencia de información válida.

Una vez ésta última preparada, es emitida hacia la unidad central, aplicándose idénticos conceptos en cuanto a estabilización de los niveles de las vías de datos, a los considerados anteriormente en el caso de las salidas. Seguidamente el periférico activa la vía de validación, lo que genera la aceptación de la palabra por parte del microprocesador.

Este proceso se repite tantas veces cuantas palabras deba enviar el periférico, de tal modo que durante el período de transmisión sólo se registra actividad de la unidad central durante los instantes en que la información es validada.

Con objeto de introducir una cierta simultaneidad, en el caso de salidas, pueden prepararse la serie de palabras a transmitir en una determinada zona de memoria, de tal modo que una vez emitida la primera palabra, el programa sigue su curso. Deben incluirse en dicho programa una serie de puntos de verificación de la ocupación del periférico, de tal modo que una vez liberado de la ocupación generada por la primera palabra, es detectada dicha situación en el próximo punto de prueba, es transmitida la segunda palabra, y sigue la ejecución del programa. Este método mejora ciertas aplicaciones, pero hace sumamente engorrosa la programación.

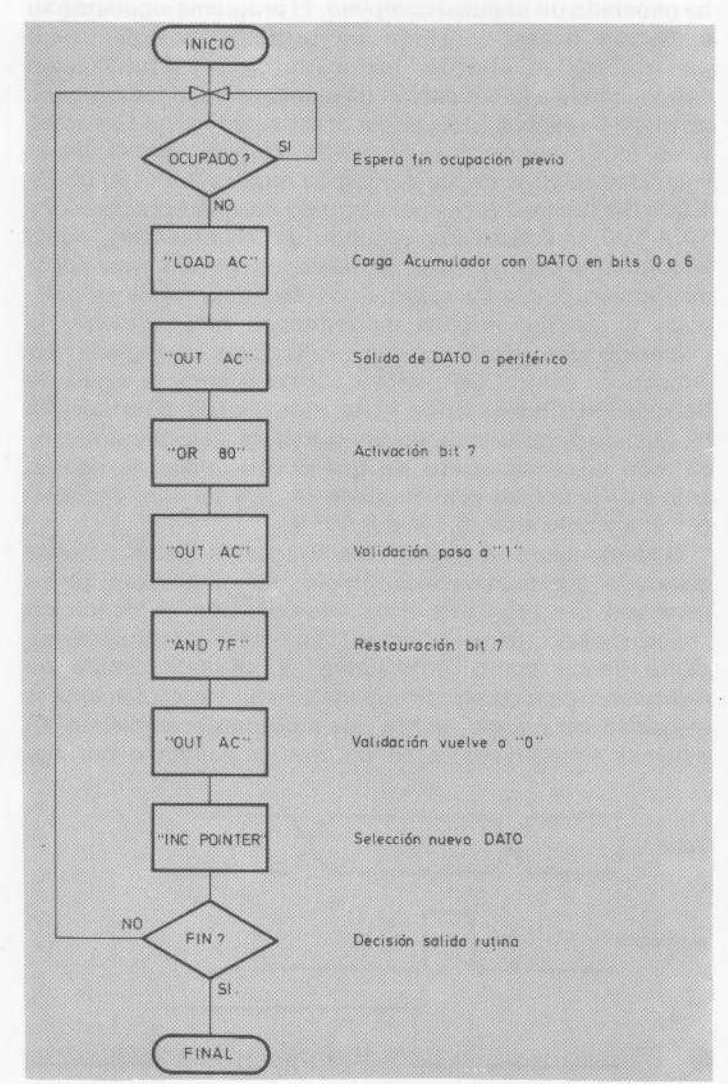


Figura 4.2 Generación de validación por programa

4.2.3 Interrupción

Otra técnica que mejora sensiblemente los tiempos de ejecución es la denominada de "interrupciones". Se denomina interrupción a la suspensión momentánea del programa en curso, memorizando los parámetros en vigor y bifurcando a una rutina para el tratamiento del periférico. Esta rutina finaliza con una instrucción de retorno al punto de suspensión del programa inicial. La suspensión del programa viene determinada por la condición de final de ejecución de la instrucción de entradas/salidas y será efectuada inmediatamente después de finalizar la instrucción que ocupe actualmente al microprocesador.

Aunque el lector ya ha sido introducido en el tema en el capítulo 1.º (§1.4.3 y siguientes), ampliamos algunos conceptos.

4.2.4 Tipos de interrupción

Algunos microprocesadores poseen un solo tipo de interrupción por lo que cuando intervenga cualquier periférico debe efectuarse en primer lugar una exploración para detectar su origen, seguidamente atender la demanda y posteriormente proseguir la ejecución normal. Inmediatamente se desprenden dos tipos de problemas, 1.º) detección del origen, y 2.º) durante el tratamiento de la interrupción, otro periférico puede a su vez interrumpir dando lugar a posibles interferencias. Para solucionar el primer problema se acude a las interrupciones vectorizadas y para el segundo al enmascaramiento y la jerarquización.

4.2.5 Interrupción vectorizada

Una interrupción se denomina vectorizada cuando simultáneamente al envío de la orden de interrupción se suministra una dirección de programa por parte del periférico, donde se encuentra la rutina específica de tratamiento. Esto se puede realizar enviando la dirección citada a través de un bus o bien disponiendo de varias entradas de interrupción en el microprocesador, cada una con una dirección asociada (a este último tipo se le denomina "restart").

4.2.6 Máscaras de interrupción

Las interrupciones pueden ser enmascarables o no. Interrupción enmascarada es aquélla que para ser activa es preciso que un biestable asociado esté en posición de autorización. Normalmente las máscaras están desactivadas autorizando la presencia de interrupciones. Al llegar una de ellas, se activan las máscaras de los periféricos o funciones, que caso de lanzar a su vez una interrupción puedan atentar contra la integridad de la función en curso de realización. Al finalizar su tratamiento se autorizan nuevamente las interrupciones, de tal modo que si en el intermedio se había presentado alguna otra, permanecería enmascarada y es ahora cuando al salir a la luz cobra actividad.

Una vez atendidas las posibles interrupciones, se devuelve el control al programa principal que sigue su curso.

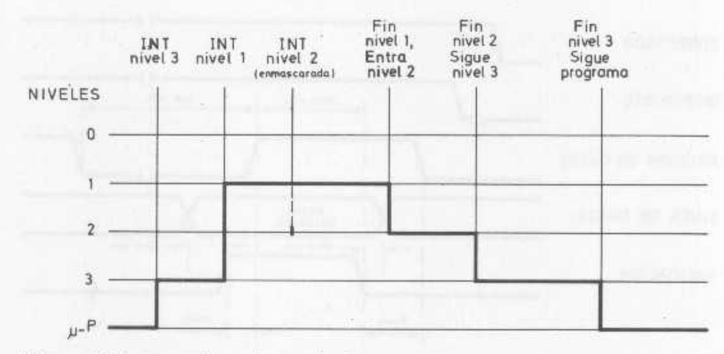


Figura 4.3 Interrupciones jerarquizadas

4.2.7 Prioridad de interrupción

Interrupciones jerarquizadas o prioritarias son aquéllas que están clasificadas por orden de importancia, de tal manera que un programa de interrupción puede ser a su vez interrumpido por otro de mayor nivel, con lo que el estado de la rutina interrumpida pasa a ser memorizado del mismo modo que lo fue el programa principal, y se pasa a ejecutar la última llegada (fig. 4.3).

4.2.8 Robo de ciclo

Cuando las características del periférico exijan una elevada velocidad de transmisión, no pudiendo ser aceptable el tiempo que media entre la solicitud de interrupción y la transmisión efectiva de la información (con la ejecución intermedia de las instrucciones precisas) se acude a la técnica de transferencia por robos de ciclo (en inglés: DMA "direct memory access": acceso directo a memoria), en este caso la secuencia de acontecimientos es la siguiente (fig. 4.4):

1.º) El programa inicia la operación, seleccionando al periférico e instruyéndole acerca del tipo de función a realizar, situación de la información en memoria, número de palabras a transmitir, etc.

2.º) El periférico emite una solicitud para que le sea concedido un ciclo de memoria, el cual le será cedido síncronamente con el próximo impulso de reloj.

3.º) El periférico direcciona la memoria y lee o deposita la primera palabra del bloque en curso de transferencia

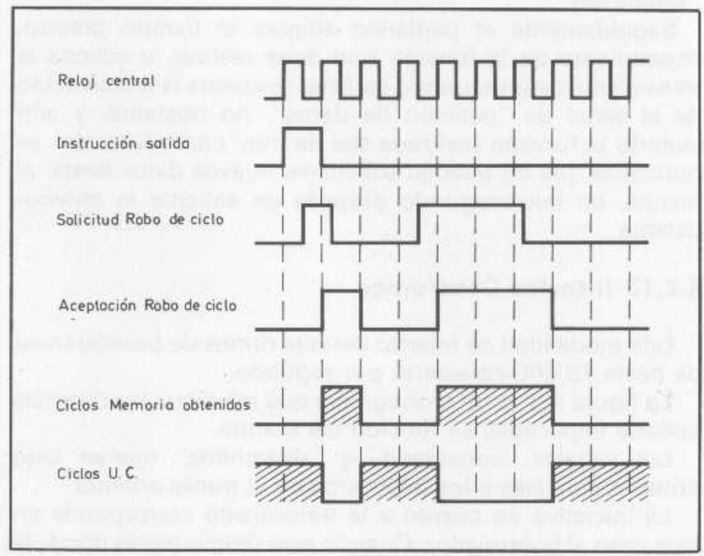


Figura 4.4 Transferencia por robos de ciclo (DMA)

mientras que el microprocesador espera a proseguir su programa al final del ciclo que le ha sido robado.

4.º) El periférico procesa la palabra durante el tiempo en que sea preciso, incrementa la dirección de memoria y solicita nuevos robos de ciclo.

5.º) Cuando el periférico completa el bloque de información, avisa al microprocesador, generalmente por medio de una interrupción o de un bus de ocupación. La rutina de interrupción podrá entonces averiguar la situación presente del periférico.

El método de robos de ciclo es evidentemente mucho más rápido, el sincronismo de operación entre la transmisión de datos y la acción del periférico es perfecta, y el retardo introducido en el programa principal es mínimo; como contrapartida exige el montaje de una electrónica más o menos compleja en cada periférico asociado.

De modo similar a las interrupciones, se suele establecer un sistema de prioridades para solventar las situaciones de simultaneidad de solicitud de robo de ciclo.

4.2.9 Encuesta

Otra modalidad de conexión de periféricos la constituye la denominada encuesta (polling) en la cual el microprocesador pregunta secuencialmente a todos los periféricos si tienen datos preparados para transmitir o bien si están en condiciones de recepción, de tal modo que los periféricos solamente están autorizados a transmitir como respuesta a consultas de la unidad central.

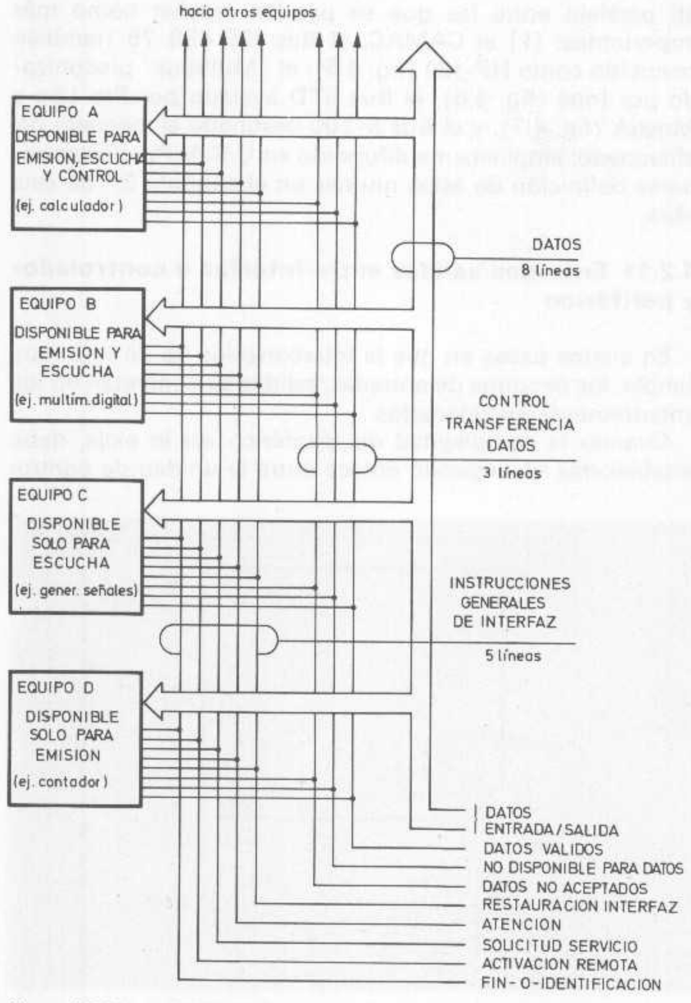


Figura 4.5 El bus IEEE-488 dispone de 8 vías de datos y 8 de control

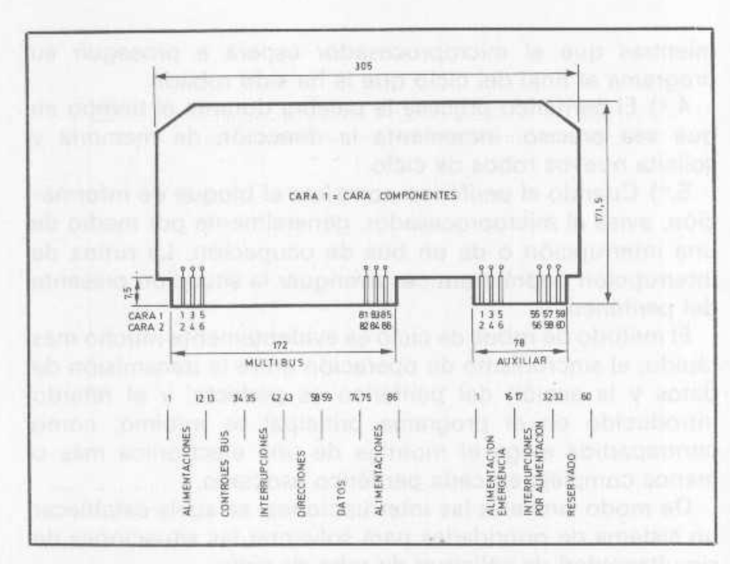


Figura 4.6 Estructura general placa y conectores MULTIBUS

4.2.10 Buses normalizados

Dada la diversidad de criterios existentes en cuanto a los protocolos de entradas/salidas, se ha presentado la necesidad de estandarizar las interfaces de tal modo que productos de distintos fabricantes puedan ser compatibles tanto a nivel de funciones, como de niveles eléctricos, como de conectores y su distribución.

Actualmente existen varias normas para la entrada/salida en paralelo entre las que se pueden reseñar como más importantes: [1] el CAMAC, el Bus IEE 488-75 (también conocido como HP-IB) (fig. 4.5); el "Multibus" preconizado por Intel (fig. 4.6), el Bus STD avalado por Pro-Log y Mostek (fig. 4.7), y el Bus S-100 destinado al mercado del aficionado, ampliamente difundido en U.S.A. Puede encontrarse definición de estas normas en el capítulo 3.º de esta obra.

4.2.11 Entradas/salidas entre interfaz o controlador y periférico

En ciertos casos en que la interconexión es de tipo muy simple, los circuitos de entradas/salidas se concretan en los anteriormente mencionados.

Cuando la complejidad del periférico así lo exija, debe establecerse un segundo enlace entre la unidad de control

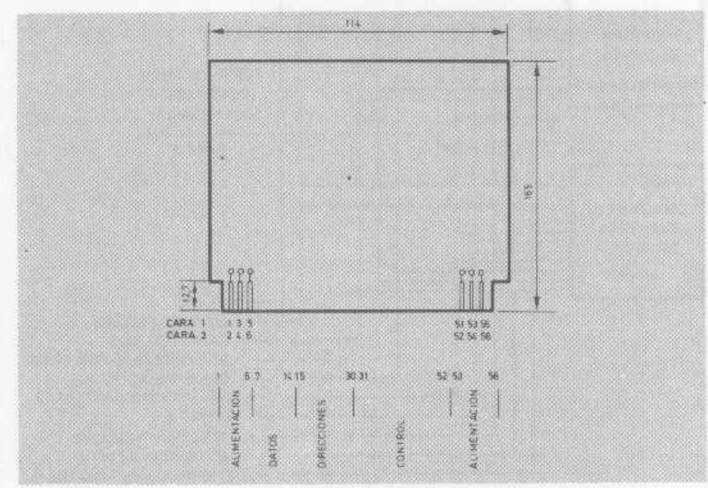


Figura 4.7 Estructura general placa y conector Bus STD

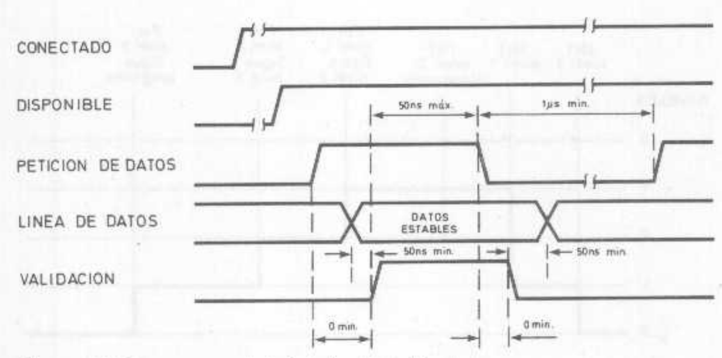


Figura 4.8 Cronograma de la interfaz Data Products

intermedia y los circuitos del propio periférico en cuestión.

Existe una gran diversidad de protocolos de interconexión depediendo del tipo de periférico y del fabricante que lo ha diseñado, por lo que se citarán a título de ejemplo dos de las interfaces de mayor difusión, fundamentalmente empleadas en periféricos de salida, y más concretamente en impresoras; aún cuando son identificadas por el nombre del fabricante que las desarrolló, son empleadas ampliamente por muy diversos suministradores.

4.2.12 Interfaz Data Products

Este tipo de interconexión permite transferencias de datos hasta ritmos equivalentes a 500.000 caracteres por segundo.

La figura 4.8 ilustra las relaciones de tiempos entre las distintas señales intercambiadas en la transferencia de datos asociada a un carácter.

Las líneas "conectado" y "disponible" son enviadas por el periférico, informando que la alimentación está conectada, el soporte (papel, cinta, etc.) está colocado, las tapas protectoras están cerradas, el conmutador "local/línea" está en posición "línea", no existe autodetección de fallos, etc.

La señal "petición de datos" es enviada asimismo por el periférico para sincronizar la transmisión de datos; como respuesta, la interfaz o controlador envía los datos por sus vías apropiadas, y con posterioridad activa la señal de "validación". Una vez el periférico detecta esta última, desconecta la señal "petición de datos" que había estado activa hasta este momento. Esta última acción, avisa al controlador que el periférico ha aceptado los datos transmitidos, por lo que puede proceder a desactivar la señal de "validación".

Seguidamente el periférico emplea el tiempo preciso, dependiente de la función que deba realizar, y solicita la transmisión de un segundo carácter mediante la reactivación de la señal de "petición de datos"; no obstante, y aún cuando la función realizada sea de muy corta duración, se normaliza que no pueden solicitarse nuevos datos hasta, al menos, un microsegundo después de solicitar la anterior palabra.

4.2.13 Interfaz Centronics

Esta modalidad de interfaz permite ritmos de transferencia de hasta 75.000 caracteres por segundo.

La figura 4.9 es un cronograma que relaciona las distintas señales implicadas en función del tiempo.

Las señales "conectado" y "disponible" operan bajo criterios similares a los expuestos en el punto anterior.

La iniciativa en cuanto a la transmisión corresponde en este caso al controlador. Cuando este último desea iniciar la comunicación, establece los niveles apropiados en las líneas

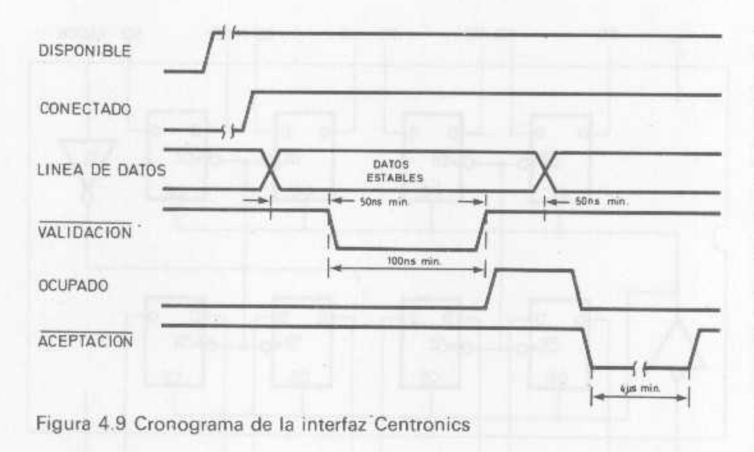


Figura 4.9 Cronograma de la interfaz Centronics

de datos, y posteriormente activa la señal de "validación".

Al finalizar la "validación", el periférico activa la señal de "ocupado" durante el tiempo preciso para ejecutar la función que le haya sido encomendada, por lo que su duración es muy variable, dependiendo específicamente de la instrucción encomendada.

Al término de su función, el periférico comunica su situación de disponibilidad para la aceptación de nuevos datos, mediante la activación de la señal de "aceptación", que asimismo confirma la correcta finalización de la orden anterior. Al término de la señal "aceptación", el controlador puede proceder a emitir una nueva palabra, repitiéndose el proceso anterior.

4.3 ELEMENTOS ASOCIADOS A BUSES

4.3.1 Microprocesadores

El elemento determinante del protocolo, tiempos de respuesta, amplitud en número de bits de la palabra transmitida, número de periféricos, etc., es la unidad central de proceso, de tal modo que para un mismo periférico, las interfaces a distintos microprocesadores son distintas no solamente a nivel electrónico, sino también en cuanto a las secuencias de programa que deben establecerse para obtener la necesaria coordinación del conjunto.

La variedad de tipos es enorme (en un directorio publicado recientemente se registran 32 microprocesadores, 45 microcomputadores y 12 "bit-slice" distintos, sin considerar los productos de proveedores alternativos) con palabras compuestas por 1, 4, 8 ó 16 bits, frecuencia máxima aceptable de reloj entre 120 kHz para los más lentos y 15 MHz para los más rápidos, tiempos de ejecución de instrucciones entre 30 ns y 780 μs, etc.

Desgraciadamente no existe ninguna normalización hasta el extremo que frecuentemente distintos productos de un mismo fabricante son totalmente incompatibles.

El único punto en que generalmente coinciden la mayoría de fabricantes es en los niveles lógicos de las señales de entradas/salidas que suelen ser compatibles con TTL (aunque existe gran cantidad de excepciones).

4.3.2 Amplificadores de bus

Son circuitos que permiten expandir la carga admisible ("fan-out") de los buses, respetando la polaridad de las señales o invirtiéndola. Las salidas generalmente son de 3 estados, es decir que una determinada señal de control cuando es activa permite que las salidas adopten los valores

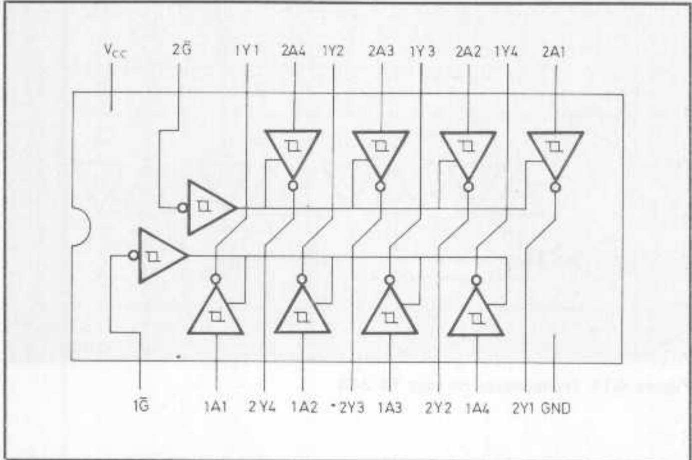


Figura 4.10 Amplificador de bus 74LS 240

binarios "0" o "1" con baja impedancia, dependiendo de los estados de las entradas; pero cuando la citada señal de control se desactiva, las salidas pasan a un tercer estado equivalente a la desconexión, presentando una elevada impedancia en serie, al margen de los estados lógicos de las entradas.

Ejemplos típicos pueden ser los 74LS240, 241 y 244 (fig. 4.10), amplificadores de bus de 8 bits de 3 estados que requieren una corriente de entrada máxima de 0,2 mA presentando características de histéresis para mejorar la susceptibilidad a ruidos, y que a su salida pueden entregar hasta 24 mA (aptos para mandar hasta 15 entradas TTL o 60 TTL LS). La distinción entre estos 3 circuitos está en el carácter inversor o no de sus entradas, de tal manera que el 240 invierte el bus y activa las salidas cuando las entradas de control son "0"; el 241 no invierte el bus y activa las salidas con señal de control "1"; y el 244 no invierte el bus y activa las salidas con señal de control "0".

4.3.3 Transceptores

Los circuitos transceptores tienen una cierta similitud con los amplificadores de bus, de los que se distinguen por el hecho de poder amplificar las señales bidireccionalmente, lo que simplifica la conexión entre la mayoría de microprocesadores cuyo bus de datos es bidireccional y periféricos con características de diálogo.

Estos circuitos presentan generalmente las salidas con características de 3 estados y disponen de señales de control que permiten habilitar la amplificación en uno u otro sentido, bien sea por señales de control de salida, independientes para cada sentido, o bien por una señal que determina el sentido y otra que autoriza las salidas.

Ejemplos típicos son 74LS242 y 243 (fig. 4.11), transceptores de 4 bits con autorización de salida en un sentido por la señal GBA en "1" y en el otro por GBA en "0". Se distinguen ambos por el hecho de que el 242 invierte el bus, mientras que el 243 respeta polaridad.

Otro ejemplo clásico son los circuitos 8216 y 8226 (fig. 4.12), asimismo transceptores de bus de 4 bits que permiten conexión directa a circuitos MOS por presentar un nivel de tensión a su salida en estado "1" como mínimo de 3,65 V. Presentan la peculiaridad de tener diferenciadas entrada y salida por un extremo, de tal modo que con conexión externa actúan como transceptores, y sin conexión entre ambas se convierten en doble amplificador de bus seleccionable [2].

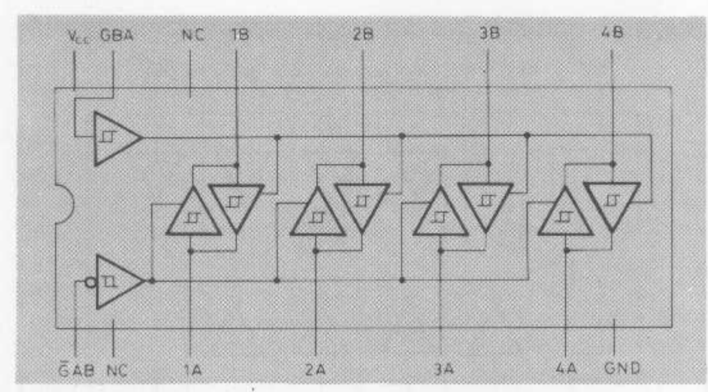


Figura 4.11 Transceptor de bus 74 243

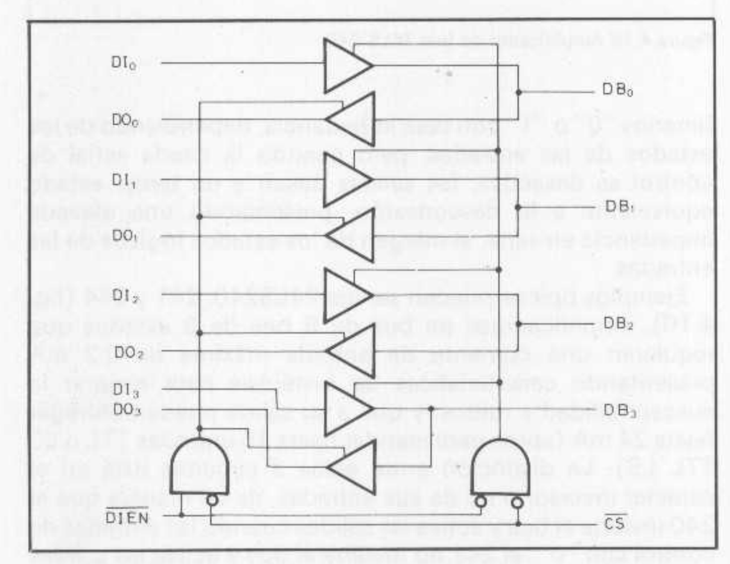


Figura 4.12 Transceptor de bus 8216

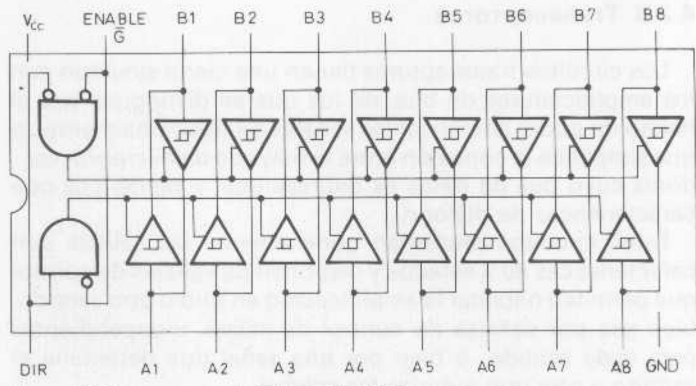


Figura 4.13 Transceptor de 8 bits 74LS 245

Como muestra de transceptor de bus de 8 bits se puede tomar el 74LS245 (fig. 4.13) en el que el control de salidas se obtiene por la combinación de una señal de autorización y otra de determinación de la dirección de la señal en el bus.

4.3.4 Registros

Un elemento de circuito muy utilizado para sincronización y memorización de datos de entrada/salida son los registros de 8 bits formados por 8 biestables tipo R-S o tipo D, con una entrada común de precarga y otra también común de autorización de las salidas de 3 estados.

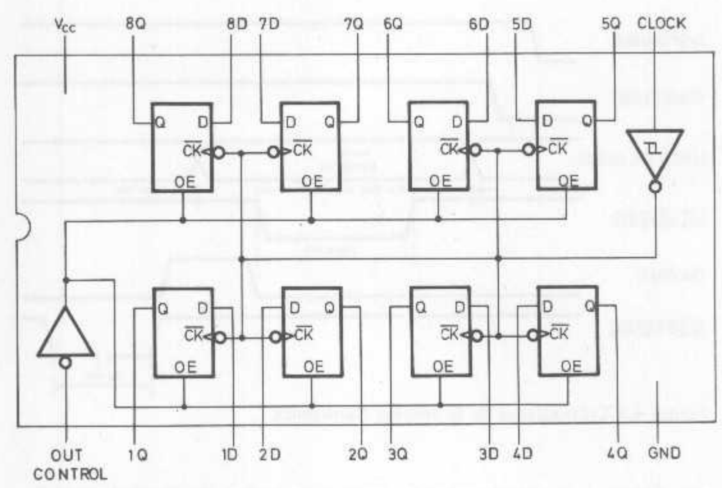


Figura 4.14 Registro octal 74LS 373

Existe una gran cantidad de modelos distintos en el mercado, indicando como referencia los siguientes: 74273, 74363, 74364, 74373, 74374 y 74377 (fig. 4.14)

4.3.5 Transceptores-registros

Un tipo particular de los antes mencionados son los transceptores-registros consistentes en un doble conjunto de 4 biestables bidireccionales con salidas de 3 estados.

Este circuito se comercializa con la denominación 74S226 (fig. 4.15) y permite transferencia bidireccional desde y hacia dos buses; asimismo la doble memorización y salida pueden ser utilizadas para realizar el intercambio de datos entre ambos buses en el tiempo equivalente de un único impulso de reloj. El almacenamiento de datos se obtiene seleccionando la función de memorización, entrando los datos, y poniendo la señal adecuada de validación a nivel "0". Mientras la validación se mantenga a nivel de cero, los datos son almacenados para la función seleccionada [3].

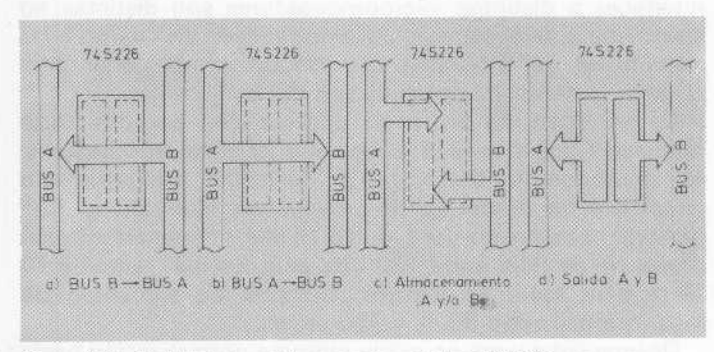


Figura 4.15 Posibilidades del transceptor-registro 74S226

Se ofrecen otros modos de actuación por medio de la posibilidad de utilizar controles de salida independientes, que pueden usarse para autorizar o desconectar las salidas según se indica en la tabla de funciones, independientemente de la función de memorización que se haya seleccionado. Las funciones de memorización pueden realizarse con las salidas seleccionadas al estado de alta impedancia. En dicho estado de alta impedancia, las entradas/salidas no ofrecen carga, ni entregan señal a los buses de modo significativo. Asimismo dispone de entradas P-N-P lo que permite una histéresis típica de 400 milivoltios para mejorar la inmunidad a ruidos.

ENTRADAS									SALIDAS			
1	2	3	4	5	6	7	8	9	D	C	B	A
H	H	H	H	H	H	H	H	H	H	H	H	H
X	X	X	X	X	X	X	X	L	L	H	H	L
X	X	X	X	X	X	X	L	H	L	H	H	H
X	X	X	X	X	X	L	H	H	H	L	L	L
X	X	X	X	X	L	H	H	H	H	L	L	H
X	X	X	X	L	H	H	H	H	H	L	H	L
X	X	X	L	H	H	H	H	H	H	L	H	H
X	X	L	H	H	H	H	H	H	H	H	L	L
X	L	H	H	H	H	H	H	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	L

H= Nivel lógico "1" L= Nivel lógico "0" X= Indiferente

Figura 4.16 Tabla de verdad del priorizador 74147

4.3.6 Priorizadores de interrupciones

Los sistemas de microprocesadores en que se utilizan interrupciones vectorizadas o jerarquizadas, están previstos generalmente para recibir información binaria respecto al elemento periférico que genera la interrupción. Dado que los periféricos, por su propia autonomía, no están relacionados entre sí "a priori", es preciso intercalar los circuitos apropiados que realicen la codificación idónea al caso. Cualquier circuito codificador podría ser apto en primera instancia, pero se presentan rápidamente dos tipos de problemáticas, si simultáneamente dos o más periféricos efectúan solicitud de interrupción. Primeramente debe impedirse que, ante pluralidad de solicitudes, puede realizarse una mezcla de bits que genere una codificación (generalmente suma binaria inclusiva) que no sea correspondiente a ninguno de los periféricos demandantes; por otro lado puede ser imperativo el establecimiento de prioridades, de tal modo que ante solicitud múltiple simultánea se establezca un arbitraje determinante de a quien se atribuye la concesión en primera instancia.

Ambas funciones pueden ser resueltas correctamente mediante los codificadores prioritarios integrados, ejemplos de los cuales son los circuitos 74147, 74148, 74278 y 74LS348 [4].

Estos codificadores TTL ofrecen una codificación prioritaria de las entradas de tal modo que puede asegurarse que solamente la línea de orden más alto pueda ser codificada. El 147 codifica 9 vías BCD en 4 bits (fig. 4.16). La condición de la línea de valor cero es implícita, por cuanto dicha condición adquiere valor positivo cuando ninguna de las otras vías es activa. Los circuitos 74148 y 74LS348 (fig. 4.17) codifican 8 líneas a 3 bits binarios, disponiendo de posibilidad de conexión en cascada sin necesidad de circuito externo mediante las señales EI (Enable Input = entrada de autorización) y EO (Enable Output = salida de

ENTRADAS									SALIDAS				
EI	0	1	2	3	4	5	6	7	A2	A1	A0	GS	EO
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	L	H	L	H
L	X	X	X	X	X	L	H	H	L	H	L	L	H
L	X	X	X	X	L	H	H	H	L	H	H	L	H
L	X	X	X	L	H	H	H	H	H	L	L	L	H
L	X	X	L	H	H	H	H	H	H	L	H	L	H
L	X	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H

Figura 4.17 Tabla funcional del codificador prioritario octal 74148

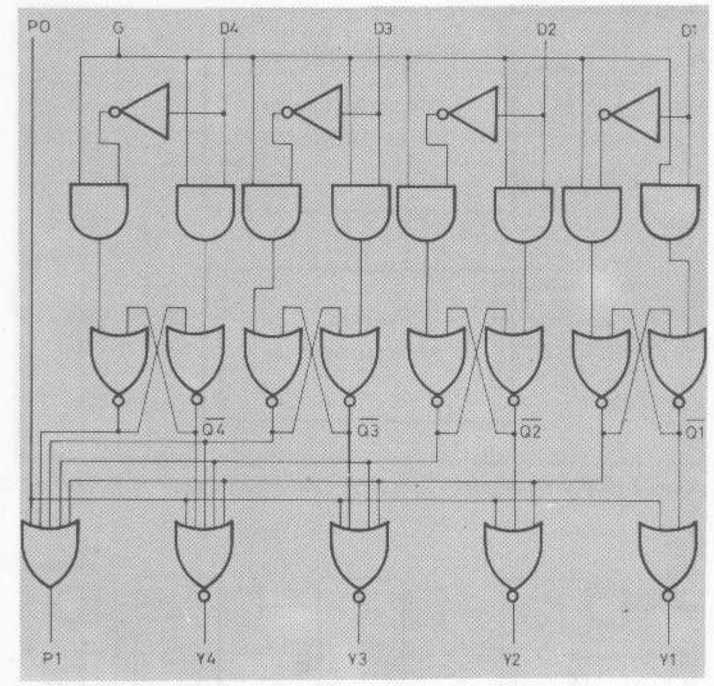


Figura 4.18 Registro prioritario 74 278

autorización), pudiendo obtener de tal modo una fácil expansión. Se puede comprender fácilmente el funcionamiento de dichos circuitos sin más que analizar las tablas de verdad asociadas.

El circuito 74278 (fig. 4.18) es un registro prioritario de 4 bits sin codificación, de funcionamiento similar a los circuitos comentados anteriormente, con la salvedad de ofrecer la facultad de memorización de órdenes suministradas en forma de impulsos. Este circuito ofrece asimismo la posibilidad de encadenamiento mediante las señales P0 y P1. La tabla de verdad es suficientemente explícita en cuanto a su modo de operación.

4.3.7 Colas de espera

A menudo se presentan problemas de sincronización motivados por distintas velocidades de transmisión y aceptación tanto en entradas como en salidas. En dichas situaciones se simplifica grandemente la problemática mediante registros de desplazamiento con secuenciamiento de colas de espera (en inglés FIFO, First-In, First-Out = primero que entra, primero que sale); estos circuitos consisten en un símil de registros de desplazamiento (shift-registers) con la particularidad de disponer de dos relojes distintos para la carga y la descarga de la información almacenada transitoriamente, con lo que pueden acondicionarse ritmos variados de transferencia entre unidades centrales y unidades periféricas.

Un ejemplo entre muchos puede ser el circuito integrado 74S225 (fig. 4.19). Este circuito es una memoria de tratamiento de colas de 16 palabras de 5 bits. Dispone de 3 señales de control. La señal "entrada disponible" indica el estado de ocupación de la última palabra, indicando cuando está llena la memoria. Esta salida está a nivel "1" cuando la memoria está disponible para aceptar nuevos datos. La salida de reloj emite un impulso en lógica negativa, sincronizado con el reloj interno, mientras dicha última posición está libre, facilitando el encadenamiento de etapas. El tercer terminal, salida disponible, se mantiene a nivel lógico "1" mientras la primera posición de memoria

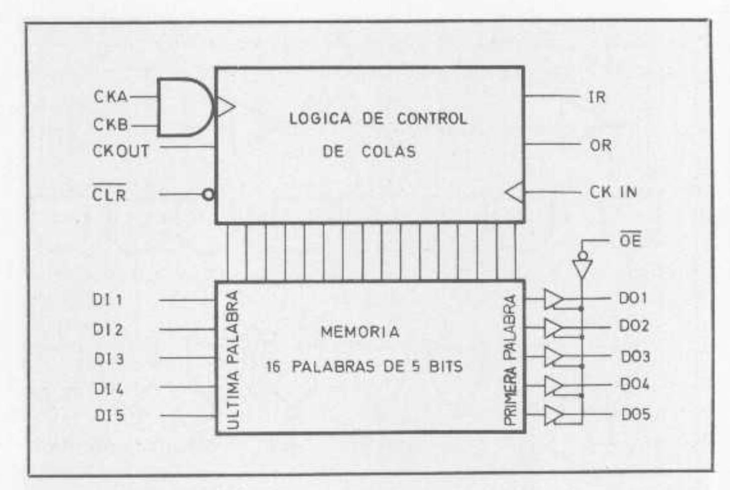


Figura 4.19 FIFO 74S225

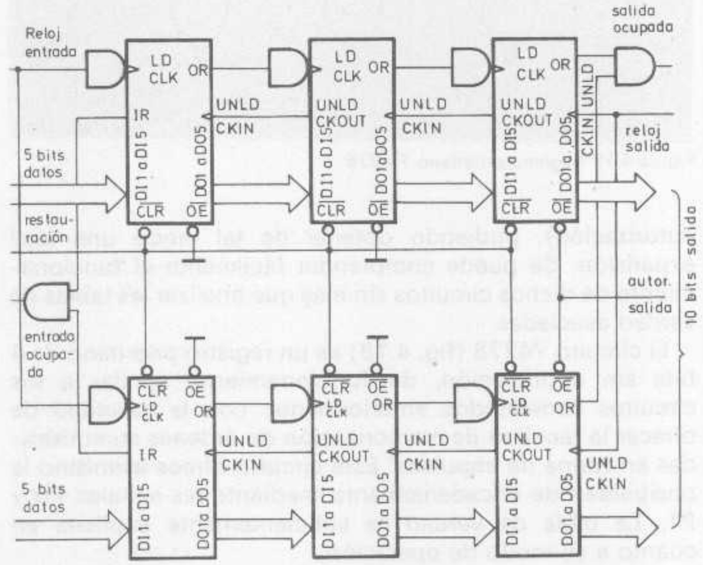


Figura 4.20 Encadenamiento de FIFOS (48 palabras de 10 bits)

contenga datos válidos y la entrada de reloj de descarga esté a nivel lógico "1". Cuando la entrada de reloj de descarga pasa a nivel "0"; la señal de disponibilidad de salida también pasa a "0". Se define la primera posición de memoria como aquélla desde la cual se dispone de datos a la salida [3].

Las salidas de datos respetan la polaridad de las entradas y son de 3 estados, con una entrada común de control. Cuando dicha señal de control está a nivel lógico «0», las salidas de datos son permitidas funcionando como salidas activas de baja impedancia. Un nivel lógico «1» fuerza las salidas a su estado de alta impedancia, mientras el resto de entradas y salidas continúan siendo activas.

Dispone asimismo de una entrada de restauración que invalida todos los datos almacenados en la memoria, poniendo a cero la lógica de control.

El encadenamiento de múltiples circuitos de este tipo es sencillo, permitiendo registros de colas de $16 \times N$ palabras por $5 \times M$ bits tal como se ilustra en el ejemplo de la figura 4.20.

4.3.8 Puertos de entrada/salida

Estos circuitos son un caso particular de registros de 8 bits y amplificadores de bus, con la adición de un circuito auxiliar para el tratamiento de interrupciones. Ejemplo clásico es el 8212 [2] o 74S412 (ambos equivalentes entre sí) realizados en tecnología TTL Schottky (fig. 4.21).

Dispone de 8 entradas DI-1 a DI-8 que cargan los 8 biestables cuando las entradas de selección DS1 y DS2 adoptan valores activos y la señal de modalidad de entrada/salida está a nivel «1», o bien cuando dicha señal es «0» y se activa la validación STB. Los estados de los 8 biestables estarán disponibles en las salidas cuando sean activas simultáneamente las señales de selección DS1 y DS2, o bien cuando la señal MD (modalidad de entrada/salida) esté a nivel lógico «1».

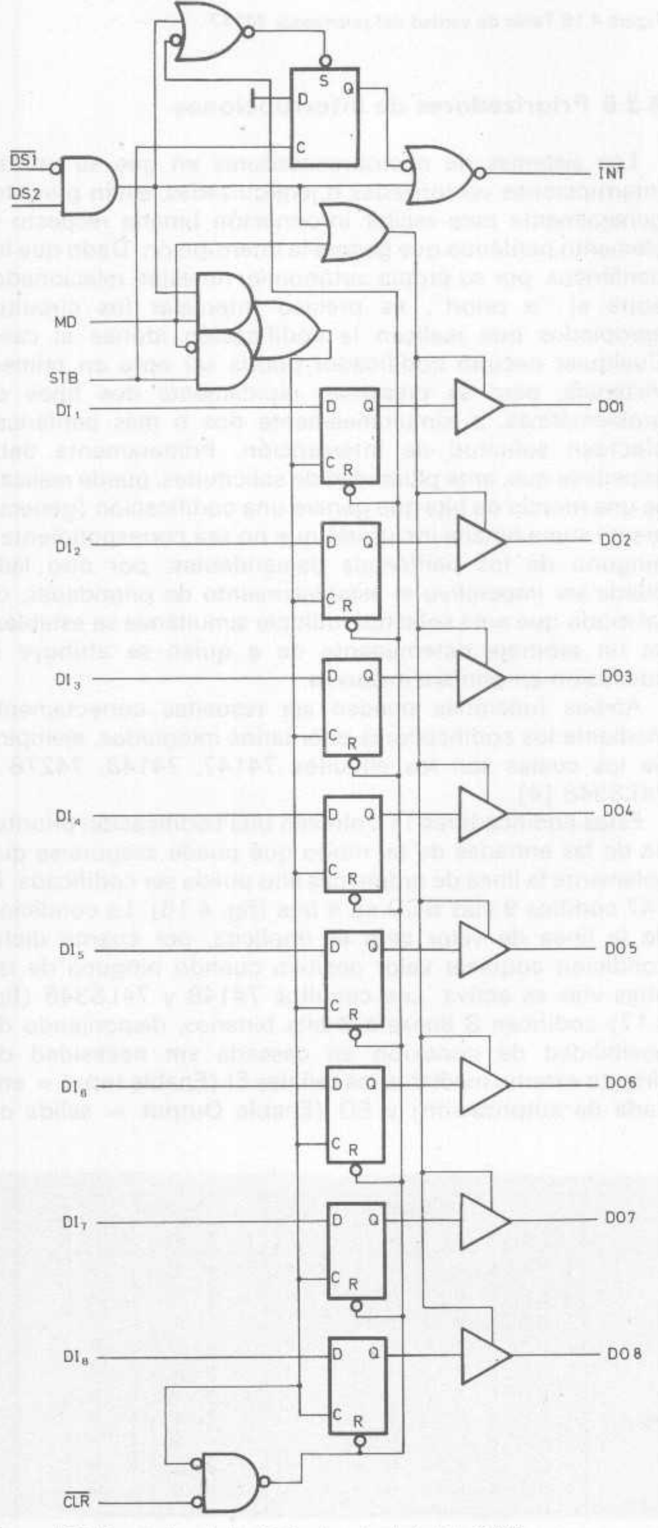


Figura 4.21 Puerto (port) de 8 bits de entrada/salida 8212

Dispone de un biestable tipo D que entrega señal de «0» a la salida INT cuando las entradas de selección son activas y se recibe una señal de validación STB; esta señal de estado puede utilizarse para indicar que el registro está cargado o bien para iniciar una secuencia de interrupción. Una entrada auxiliar CLR restaura a «0» todos los bits simultáneamente al ser activada.

4.3.9 Controladores de entradas/salidas en paralelo

Los fabricantes de microprocesadores han desarrollado múltiples circuitos específicamente desarrollados para el control de entradas/salidas en paralelo dependientes de sus propias unidades centrales, seguidamente se describen los circuitos más representativos.

4.3.10 Interfaz programable 8255A

El 8255A es un circuito programable de entradas/salidas de utilización general (fig. 4.22) diseñado originalmente para ser acoplado a los microprocesadores INTEL 8080 y 8085. Dispone de 24 terminales de E/S que pueden ser programados individualmente en dos grupos de doce y ser empleados en tres distintas modalidades principales. En la primera modalidad (Modo 0), cada grupo de doce terminales de entradas/salidas puede ser programado en grupos de cuatro para actuar como entradas o como salidas. En Modo 1 (la segunda modalidad), cada grupo puede ser programado para disponer de 8 líneas de entrada o salida, mientras que tres de los restantes terminales, por cada grupo, son utilizados para sincronización de entradas/salidas sea por esperas o por interrupciones. La tercera modalidad de operación (Modo 2) adopta la configuración de bus bidireccional, empleando 8 líneas para datos y 5 para sincronización (tomando una del otro grupo) [2].

La configuración funcional de cada puerto es programada por el sistema. En esencia el microprocesador emite una palabra de control al 8255, conteniendo información tal como: modalidad, activación de bits, restauración de bits, etc., con el fin de inicializar el circuito.

El registro de control solamente puede ser escrito, es decir

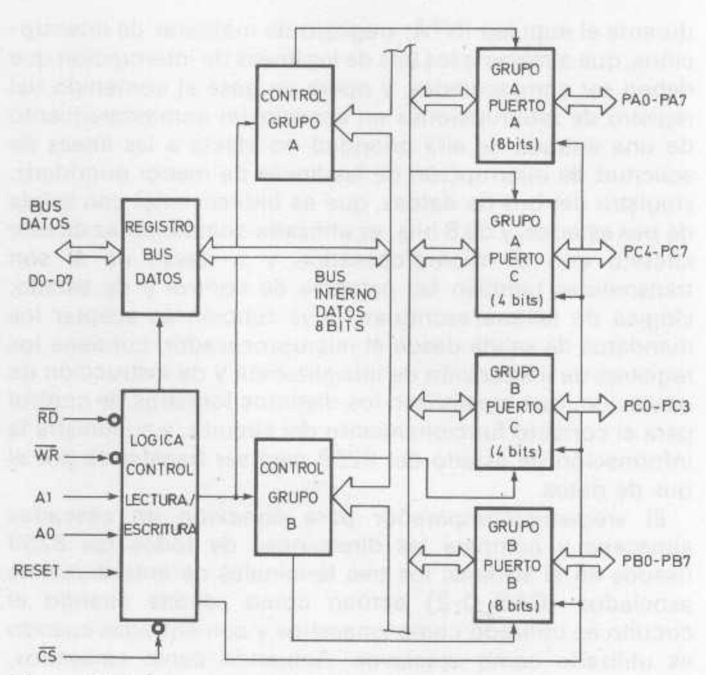


Figura 4.22 Interfaz programable 8255A

que no es posible leer su estado en un momento determinado desde el microprocesador.

4.3.11 Interfaz universal 8041/8741

Los circuitos 8041 y 8741 constituyen interfaces programables de utilización general, diseñados para ser utilizados con varios microprocesadores de 8 bits (fig. 4.23). Contienen internamente microprocesador económico con memoria de programa, unidad central de proceso de 8 bits, «ports» de entradas/salidas, un contador/temporizador, y un circuito generador de reloj, todo ello en una cápsula de 40 terminales. Inicialmente se comercializa como controlador de periféricos para los sitemas INTEL MCS-80, MCS-85 y MCS-48, pero puede adaptarse a otros microprocesadores de 8 bits [2].

Este circuito dispone de 1 K palabras de memoria de programa y 64 palabras de memoria de datos, ambas internas. Para permitir mayor flexibilidad, la memoria de programa existe en dos versiones: el 8041 la tiene en versión ROM de máscara y el 8741 en forma de PROM borrable por radiaciones ultravioletas. Ambos son totalmente compatibles, de tal modo que el 8741 está enfocado hacia desarrollo de prototipos y pequeñas series, pero cuando el volumen de producción lo hace interesante se puede pasar al 8041 sin modificación alguna en el circuito.

Los 8041/8741 disponen de 2 «ports» de 8 bits cada uno, bidireccionales y compatibles TTL y dos entradas de prueba. Individualmente, las líneas de los «ports» pueden actuar como entradas o como salidas, bajo control del programa. Se incluye un temporizador/contador programable de 8 bits con objeto de generar secuencias cronometradas o bien contar impulsos externos aplicados a las entradas. Otras facetas interesantes son: alimentación única a 5 V; bajo consumo en estado de latencia (en la versión 8041); posibilidad de efectuar el programa paso a paso, facilitando su desarrollo y análisis (en la versión 8741); un nivel de interrupciones, y dos bancos de registros de trabajo.

Dado que constituye un microprocesador completo, permite mayor flexibilidad que los circuitos convencionales LSI para interfaz. Está diseñado para ser un controlador eficiente, así como un procesador aritmético.

4.3.12 Controlador de interrupciones programable (8259)

El circuito 8259 desarrollado por INTEL para sus microprocesadores 8080 y 8085 permite controlar hasta 8 interrupciones prioritarias vectorizadas (fig. 4.24). Puede ser acoplado en cascada, para ampliar el sistema hasta 64 interrupciones sin ningún circuito adicional.

Este circuito es estático y no precisa entrada de reloj externo [2]. El funcionamiento general puede ser estudiado a partir de su esquema de bloques. Los bloques y señales más importantes son los siguientes: «registro de interrupciones solicitadas» y «registro de interrupciones en servicio», en los cuales las entradas de interrupción por las líneas IR son controladas por dos registros en cascada, el registro de interrupciones solicitadas, usado para almacenar todos los niveles de interrupción que solicitan atención y el registro de interrupciones en servicio que almacena los niveles de interrupción que están siendo atendidos; «priorizador», bloque lógico que determina las prioridades de los bits seleccionados en el registro de solicitud de interrupciones, la prioridad de mayor nivel es la seleccionada y validada en el correspondiente bit del registro de interrupciones en servicio

Figura 4.23 Interfaz universal 8041

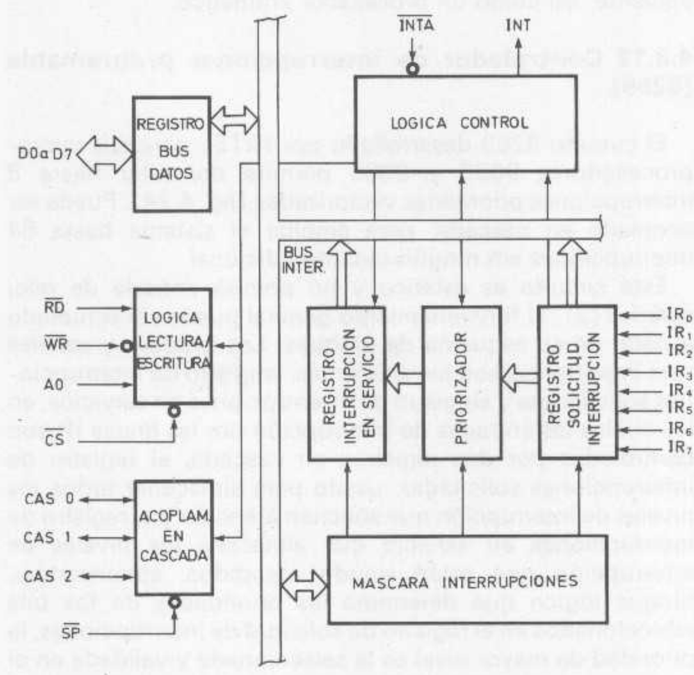


Figura 4.24 Controlador de interrupciones programable 8259

durante el impulso INTA; «registro de máscaras de interrupción», que almacena los bits de las líneas de interrupción que deben ser enmascaradas, y opera en base al contenido del registro de interrupciones en servicio (el enmascaramiento de una entrada de alta prioridad no afecta a las líneas de solicitud de interrupción de las líneas de menor prioridad); «registro del bus de datos», que es bidireccional con salida de tres estados, y de 8 bits, es utilizado como interfaz de este circuito con el microprocesador, y a través de él son transmitidas también las palabras de control y de estado; «lógica de lectura/escritura», cuya función es aceptar los mandatos de salida desde el microprocesador, contiene los registros de instrucción de inicialización y de instrucción de operación, que memorizan los distintos formatos de control para el correcto funcionamiento del circuito, y suministra la información de estado del 8259 para ser transferida por el bus de datos.

El «registro/comparador para conexión en cascada» almacena y compara las direcciones de todos los 8259 usados en el sistema; los tres terminales de entrada/salida asociados (CAS 0-2) actúan como salidas cuando el circuito es utilizado como «maestro» y son entradas cuando es utilizado como «esclavo». Actuando como «maestro», envía a la dirección del circuito «esclavo» que puede

interrumpir por las líneas CAS 0-2; el «esclavo» seleccionado de este modo enviará su subrutina programada por el bus de datos durante los 2 próximos impulsos INTA.

En el capítulo 6.°, apartado 6.6 se trata y describe nuevamente este controlador.

4.3.13 Controlador de acceso directo a memoria programable (8257)

El 8257 es un circuito controlador de acceso directo a memoria programable que, acoplado a un puerto 8212, forma un completo controlador de cuatro canales para ser utilizado en sistemas de microprocesadores INTEL (fig. 4.25). Después de haber sido inicializado por programa, este circuito puede transferir un bloque de datos, conteniendo hasta 16.384 palabras entre memoria y un periférico directamente, sin otra intervención precisa por el microprocesador [2].

Tras recibir una solicitud de transferencia por robo de ciclo desde un periférico, el 8257 adquiere control del bus del sistema, reconoce y selecciona el periférico que está conectado en el canal de mayor prioridad, emite los 8 bits menos significativos en los buses de memoria hacia las líneas de direccionamiento A0-A7, mientras emite los 8 bits más significativos de la dirección de memoria hacia el puerto de entrada/salida 8212 a través del bus de datos (el 8212 sitúa estos bits de dirección en las líneas A8-A15) y genera las señales de control apropiadas para memoria y entrada/-salida que permitan las funciones tanto de lectura como de escritura de forma tal que el periférico pueda recibir o depositar un octeto de datos directamente desde o hacia la posición direccionada de memoria.

Este circuito pueda mantener el control del bus del sistema y repetir la secuencia de transferencia mientras que el periférico mantenga su solicitud de acceso directo a memoria. De este modo el 8257 puede transferir un bloque de datos hacia o desde un periférico de alta velocidad (por ejemplo un sector de datos de un disco flexible) en un único tren de impulsos. Cuando se han transferido el número apropiado de palabras de datos, el 8257 activa la señal PC, informando a la Unidad Central que la operación ha sido completada.

Figura 4.25 Controlador de acceso directo a memoria programable 8257

Asimismo este circuito permite tres modos distintos de operación: 1.°) lectura de memoria; esta función provoca que los datos sean transferidos desde memoria hasta el periférico; 2.°) escritura de memoria, que causa que los datos puedan ser transferidos desde un periférico hacia la memoria, y 3.°) verificación de memoria, que no provoca por sí misma una transferencia de datos. Cuando un 8257 genera una instrucción de verificación de memoria actúa de modo similar a como ha sido descrito para las operaciones de transferencia, excepto que no se genera ninguna señal de control de memoria ni de entrada/salida, impidiendo por tanto la transferencia de datos. En esta situación este circuito obtiene el control del bus de sistema y puede reconocer qué periférico está solicitando datos durante cada ciclo de acceso a la memoria. El periférico que provoca esta señal de reconocimiento es autorizado a ganar control del sistema, de forma tal que pueda ejecutar algunos procesos de verificación, tales como acumulación del CRC (código cíclico de redundancia). Por ejemplo, tras un bloque de ciclos de lectura de memoria se puede realizar una serie de ciclos de verificación de memoria permitiendo al periférico autentificar la calidad de los datos recientemente adquiridos.

4.3.13.1 *Descripción en bloques del sistema*

Canales de acceso directo a la memoria. Este circuito dispone de cuatro canales distintos (rotulados CH-0 hasta CH-3). Cada canal incluye dos registros de 16 bits: 1.°) un registro de dirección de memoria, y 2.°) un registro contador de terminal. Ambos registros deben ser inicializados cuando el canal se conecta. El registro de dirección de memoria es cargado con la dirección de la primera posición de memoria que debe ser recibida. El valor cargado en los 14 bits de menor peso del registro contador de terminal especifica el número de ciclos de memoria menos uno que se van a realizar antes de que el contaje de terminal (TC) sea activado. Por ejemplo, un contaje de terminal igual a 0 causará que la salida TC sea activa durante el primer ciclo de memoria de dicho canal. En general, si N es igual al número de ciclos de memoria deseados, se cargará el valor N-1 en los 14 bits de menor peso de dicho registro. Los dos bits de mayor peso de este registro especificarán el tipo de acceso a memoria para dicho canal según se indica en la tabla de funcionamiento. Estos dos bits no son modificados desde un ciclo de acceso a memoria pero pueden ser cambiados entre bloques de acceso.

Registro del bus de datos. Este registro de 8 bits bidireccional y de 3 estados realiza la interfaz desde este circuito hacia el bus de datos del sistema.

(D0-D7), líneas del bus de datos. Estas líneas son bidireccionales y de 3 estados. Durante el proceso de inicialización del circuito a través del microprocesador, los 8 bits de datos del registro de dirección de acceso a la memoria, el registro de contaje del terminal o el registro del modo de operación son preseleccionados y recibidos a través de este bus. Cuando la unidad central lee el registro de dirección de memoria, el registro de contaje del terminal o el registro de estado, los datos son enviados hacia la unidad central por medio del bus de datos. Durante los ciclos de

acceso a memoria (cuando el 8257 es el controlador de bus), este circuito emitirá los 8 bits más significativos de la dirección de memoria hacia el registro 8212 a través del bus de datos. Estos bits de dirección serán transmitidos al principio del ciclo de acceso; a partir de este momento el bus será liberado para permitir a la memoria efectuar la transferencia de datos durante el resto del ciclo.

Lógica de lectura/escritura. Cuando la unidad central está programando o leyendo uno de los registros internos, la lógica de lectura/escritura acepta la lectura de entrada/salida (I/OR) o la escritura de entrada/salida (I/OW), decodifica los 4 bits de dirección menos significativos (A0-A3) y asimismo transfiere el contenido del bus de datos en el registro direccionado, si la señal I/OW es activa; o bien sitúa el contenido de registro seleccionado en el bus de datos si (I/OR) es activa. Durante los ciclos de acceso a memoria, la lógica de lectura/escritura genera las señales de lectura de entrada/salida y escritura de memoria o bien la escritura de entrada/salida y lectura de memoria, de forma tal que pueda controlar el enlace de datos entre el periférico que ha solicitado el control y la memoria central. Nótese que durante las transferencias de acceso directo a memoria los circuitos que no deben realizar este tipo de transferencia deben ser desconectados utilizando la señal «AEN» inhibiendo la decodificación de estos periféricos de entrada/salida de la dirección de memoria para impedir selecciones erróneas.

(I/OR), lectura de entrada/salida. Cuando se trabaja en modo esclavo, ésta es una entrada que permite leer los 8 bits del registro de estado o la palabra alta o baja de los 16 bits del registro de dirección de memoria o bien el registro de contaje de terminal. Si se está trabajando en modo maestro, ésta es una salida de control que puede ser utilizada para acceder datos desde un periférico durante un ciclo de escritura de memoria.

(I/OW), escritura de entrada/salida. Cuando se trabaja en modo esclavo ésta es una entrada que permite cargar el contenido de bus de datos en los 8 bits del registro de modo de operación o bien cargar la palabra alta o baja de los 16 bits del registro de dirección de memoria o bien el registro de contaje de terminal. Cuando se trabaja en modo maestro esta señal es una salida de control que permite a los datos salir hacia un periférico durante el ciclo de lectura de memoria.

Lógica de control. Este bloque contiene la secuencia de

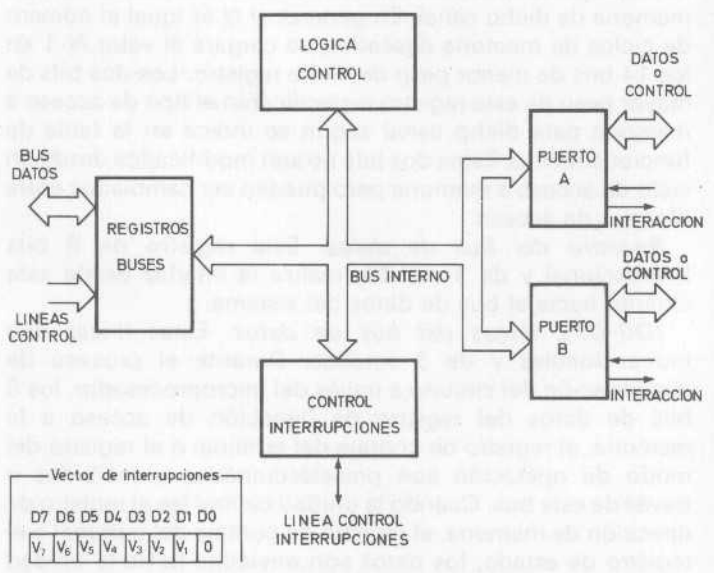


Figura 4.26 Controlador de interfaz paralelo Z-80-PIO

operaciones durante todos los ciclos de memoria generando las apropiadas señales de control y los 16 bits de dirección que especifican la posición de memoria que debe ser accedida.

4.3.14 Controlador de interfaz paralelo (Z-80-PIO)

El controlador de interfaz paralelo Z-80-PIO es un circuito programable que contiene dos puertos de entrada/salida con interfaz compatible TTL permitiendo conexión de periféricos a la unidad central Z-80-CPU (fig. 4.26). Sus características principales son las siguientes: control de generación de interrupciones interactivas para rápida respuesta; selección de cualquiera de los siguientes modos de operación por cualquier vía de acceso (port): salida de palabra, entrada de palabra, bus bidireccional de palabra (disponible sólo para la vía de acceso A), modalidad de bit; interrupciones programables en base a las condiciones de estado de los periféricos; inclusión de lógica de interrupciones prioritarias encadenadas permitiendo una interrupción vectorizada automática sin necesidad de lógica externa; sus 8 salidas son capaces de activar transistores tipo Darlington; todas las entradas y salidas son totalmente compatibles con niveles TTL.

Se muestra un diagrama de bloques de este circuito en la figura 4.26.

La estructura interna de dicho circuito consiste en un bus de interfaz con la unidad central Z-80, lógica de control interna, lógica de entrada/salida de la vía de acceso A, lógica de entrada/salida de la vía de acceso B, y lógica de control de interrupciones.

Una aplicación típica utilizará la vía de acceso A como canal de transferencia de datos y la vía de acceso B para monitorización de estados y control.

La lógica de las vías de acceso de entrada/salida está compuesta por 6 registros con lógica de control interactiva. Los registros incluyen: un registro de entrada de 8 bits, un registro de salida de 8 bits, un registro de modalidad de control de 2 bits, un registro de máscara de 8 bits, un registro de selección de entrada/salida de 8 bits, y un registro de máscaras de 2 bits. Los últimos tres registros son utilizados solamente cuando la vía de acceso (port) ha sido programada para operar en la modalidad de bit.

El registro de control de modalidad (2 bits) es cargado por la unidad central para seleccionar la modalidad de operación: salida de palabra, entrada de palabra, bus bidireccional de palabra o modalidad de bit. El registro de salida de datos (8 bits) permite que los datos sean transferidos desde la unidad central hacia el periférico. El registro de entrada de datos (8 bits) acepta datos desde el periférico para ser transferidos a la unidad central. El registro de control de máscara (2 bits) es cargado por la unidad central para especificar el estado activo (alto o bajo) de cualquier circuito periférico que deba ser controlado, o bien si debe generarse una interrupción, cuando todos los terminales no enmascarados son activos (condición Y), o cuando cualquier terminal no enmascarado está activo (condición O). El registro de máscara (8 bits) es cargado por la unidad central para determinar qué circuito periférico debe ser monitorizado para la condición específica de estado. El registro de selección de entrada/salida (8 bits) es cargado por la unidad central para permitir a cualquier terminal ser una salida o una entrada durante la operación en modalidad de bit.

Modalidad de salida. Un ciclo de salida siempre es disparado por la ejecución de una instrucción de salida por parte de la unidad central. El impulso WR desde la unidad central provoca la memorización de datos desde el bus de

datos de la unidad central hacia el registro de salida del «port» seleccionado. El impulso de escritura selecciona el indicador «libre» cuando aparezca el flanco de bajada de reloj, indicando que los datos están disponibles. La condición de «libre» permanece activa hasta recibir un flanco positivo por la línea de validación indicando que los datos han sido adquiridos por el periférico. El flanco positivo del impulso de validación genera una interrupción si el flip-flop de autorización de interrupción ha sido activado y si este circuito tiene la prioridad de orden más elevado.

Modalidad de entrada. Cuando la señal STROBE ofrece un flanco de bajada, los datos son cargados en el registro de entrada de la vía de acceso seleccionada. El próximo flanco de subida de la señal de validación activa INT si se ha activado la autorización de interrupción y si este circuito tiene la prioridad de mayor orden. El siguiente flanco de bajada del reloj restaura «libre» hacia un estado inactivo, indicando que el registro de entrada está lleno y no puede aceptar más datos hasta que la unidad central complete una lectura. Cuando se complete una lectura, el flanco positivo de RD activará «libre» (ready) en la próxima transición negativa del reloj. A partir de este momento pueden ser cargados nuevos datos en el controlador paralelo de entrada/salida.

Modalidad bidireccional. Esta es una combinación de los modos 0 y 1 utilizando las 4 líneas interactivas y las 8 líneas de la vía de acceso A. La vía de acceso B debe ser activada en modalidad de bit, las líneas interactivas de la vía de acceso A son utilizadas para control de salida y las líneas de la vía de acceso B son utilizadas para entrada de control. Se permite la salida de datos por el bus de la vía de acceso A, solamente cuando la señal A STB ha descendido a nivel «0». El flanco de subida de esta señal de validación puede ser utilizado para memorizar estos datos en el periférico.

Modalidad de bit. La modalidad de bit no utiliza las señales interactivas y permite ejecutar una escritura normal de «port» o bien una lectura, en cualquier momento. Cuando se desee una escritura, los datos serán almacenados en los registros de salida, bajo las mismas condiciones de la modalidad de salida. Cuando este circuito esté leyendo, los datos enviados a la unidad central estarán compuestos por datos del registro de salida de aquellas líneas de la vía de acceso de datos asignada como salida y por entradas de datos de aquellas líneas de datos de vía de acceso asignadas como entradas. El registro de entrada contendrá los datos que han sido presentados inmediatamente antes del flanco de bajada de RD. Puede ser generada una interrupción si están autorizadas las interrupciones desde dicha vía de acceso (port) y la información en las líneas de datos de la vía de acceso (port) satisface la ecuación lógica definida por la máscara de 8 bits y el registro de control de máscara de 2 bits.

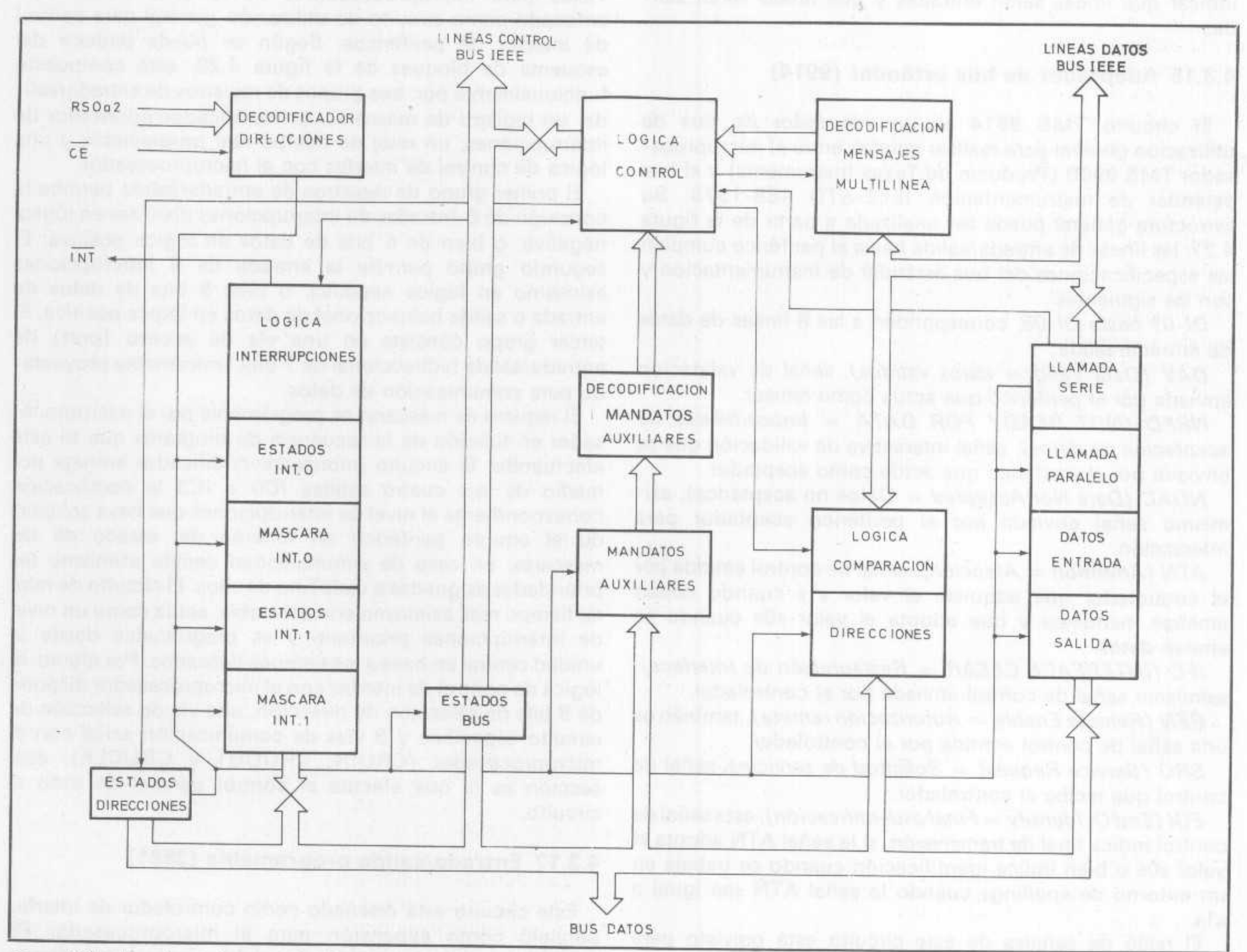


Figura 4.27 Adaptador de bus estándar 9914

Carga del vector de interrupciones. La unidad central Z-80 precisa un vector de interrupciones de 8 bits suministrado por parte del circuito interruptor. La unidad central forma la dirección de la rutina de servicio de interrupción del «port» utilizando este vector. Durante un ciclo de reconocimiento de interrupción el vector es situado en el bus de datos de la unidad central por el circuito de mayor prioridad que está seleccionando un servicio en dicho momento. El vector de interrupciones deseado puede ser cargado en dicho circuito escribiendo una palabra de control en la vía de acceso o puerto (port) deseado del Z-80-PIO con el formato indicado en la figura 4.26.

Selección de un modo de operación. Cuando se selecciona un modo de operación, el registro de control de modalidad (de 2 bits) es seleccionado a uno de sus 4 posibles valores. Estos 2 bits son los más significativos del registro (bits 7 y 6); los bits 5 y 4 no son utilizados mientras que los bits desde el 3 hasta el 0 están todos preseleccionados a «1» para indicar modo de selección. Modo 0 activo indica que los datos serán escritos desde la unidad central hacia el periférico. Modo 1 activo indica que los datos serán leídos desde el periférico hacia la unidad central. Modo 2 permite que los datos sean escritos o leídos desde la unidad periférica. Modo 3 está previsto para aplicaciones de control y estado. Cuando se selecciona, la próxima palabra de control seleccionará el registro de entrada/salida para indicar qué líneas serán entradas y qué líneas serán salidas.

4.3.15 Adaptador de bus estándar (9914)

El circuito TMS 9914 es un adaptador de bus de utilización general para realizar interfaz entre el microprocesador TMS 9900 (Producto de Texas Instruments) y el bus estándar de instrumentación IEEE-STD 488-1975. Su estructura general puede ser analizada a partir de la figura 4.27; las líneas de entrada/salida hacia el periférico cumplen las especificaciones del bus estándar de instrumentación y son las siguientes:

DI-01 hasta DI-08, corresponden a las 8 líneas de datos de entrada/salida.

DAV (Data Valid = datos válidos), señal de validación enviada por el periférico que actúa como emisor.

NRFD (NOT READY FOR DATA = Imposibilidad de aceptación de datos), señal interactiva de validación que es enviada por el periférico que actúe como aceptador.

NDAC (Data Not Accepted = Datos no aceptados), asimismo señal enviada por el periférico aceptador para interacción.

ATN (Attention = Atención), señal de control emitida por el controlador que adquiere el valor «1» cuando deban emitirse mandatos y que adopta el valor «0» cuando se emitan datos.

IFC (INTERFACE CLEAR = Restauración de Interface), asimismo señal de control enviada por el controlador.

REN (Remote Enable = Autorización remota), también es una señal de control emitida por el controlador.

SRQ (Service Request = Solicitud de servicio), señal de control que recibe el controlador.

EOI (End Or Identify = Final o identificación), esta señal de control indica final de transmisión, si la señal ATN adopta el valor «0» o bien indica identificación cuando se trabaja en un entorno de «polling» cuando la señal ATN sea igual a «1».

El resto de señales de este circuito está previsto para realizar la interfaz con la unidad central.

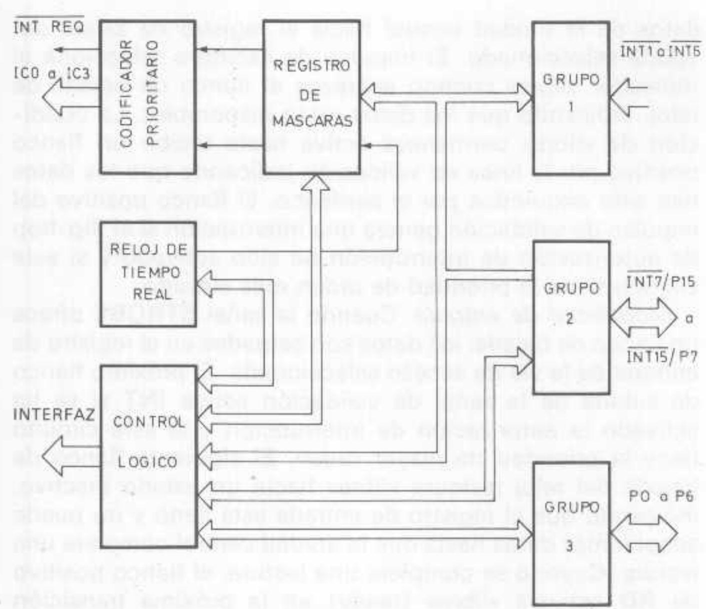


Figura 4.28 Interfaz programable 9901

4.3.16 Interfaz programable (9901)

El circuito TMS 9901 es otro componente de la familia Texas para microprocesadores de la serie 9900. Está enfocado como circuito de utilización general para control de interfaz de periféricos. Según se puede deducir del esquema de bloques de la figura 4.28, está compuesto funcionalmente por: tres grupos de registros de entrada/salida, un registro de máscaras, un codificador/priorizador de interrupciones, un reloj de tiempo real programable y una lógica de control de interfaz con el microprocesador.

El primer grupo de registros de entrada/salida permite la conexión de 6 entradas de interrupciones distintas en lógica negativa, o bien de 6 bits de datos en lógica positiva. El segundo grupo permite la entrada de 9 interrupciones asimismo en lógica negativa, o bien 9 bits de datos de entrada o salida bidireccional de datos en lógica positiva. El tercer grupo consiste en una vía de acceso (port) de entrada/salida bidireccional de 7 bits, únicamente proyectado para comunicación de datos.

El registro de máscaras es programable por el microprocesador en función de la secuencia de programa que se esté efectuando. El circuito priorizador/codificador entrega por medio de sus cuatro salidas IC0 a IC3 la codificación correspondiente al nivel de interrupciones que haya solicitado el equipo periférico en función del estado de las máscaras; en caso de simultaneidad decide asimismo las prioridades asignadas a cada uno de ellos. El circuito de reloj de tiempo real, asimismo enmascarable, actúa como un nivel de interrupciones prioritario y es programable desde la unidad central en base a los tiempos deseados. Por último, la lógica de control de interfaz con el microprocesador dispone de 5 bits de selección de dirección, una vía de selección de circuito específico y 3 vías de comunicación serial con el microprocesador (CRUIN, CRUOUT y CRUCLK); esta sección es la que efectúa el control general de todo el circuito.

4.3.17 Entrada/salida programable (3861)

Este circuito está diseñado como controlador de interfaz paralelo como expansión para el microprocesador F8 producido originalmente por Fairchild y actualmente dispo-

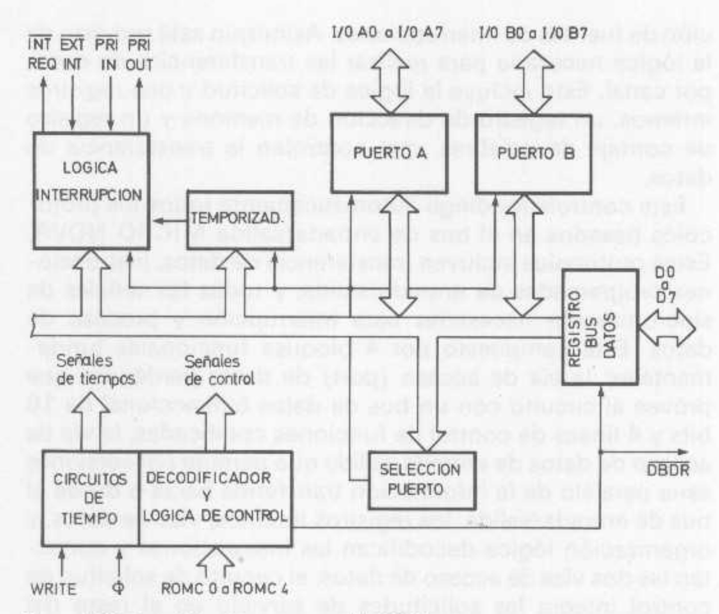


Figura 4.29 Entrada/salida programable 3861

niendo de segundo suministro a través de Mostek, Motorola y SGS-ATES. Está compuesto por dos vías de acceso de entrada/salida de 8 bits, un temporizador programable, y un temporizador vectorizado o interrupción externa aptos para el sistema F8 [5]. La figura 4.29 indica los distintos bloques funcionales y registros. Este circuito utiliza las señales de reloj «Φ» y «WRITE», ambas generadas por la unidad central, como funciones de control de tiempo dentro del circuito. También utiliza el contenido de 5 líneas de control (ROMC) como instrucciones de las funciones que debe realizar en cada ciclo. Un circuito de control formado por una ROM decodifica estas 5 líneas de control y establece las secuencias de funcionamiento.

«Ports» direccionales. Este circuito tiene 4 vías de acceso direccionales. Todas ellas están enlazadas con el acumulador de la unidad central por medio de las instrucciones de entrada/salida. Cada vía de acceso está referenciada por una dirección de 8 bits. Los 6 bits de orden más alto de la dirección se refieren al circuito en el cual las vías de acceso están situadas, mientras que los 2 bits de orden inferior seleccionan una de las 4 vías; por tanto las direcciones de vías de acceso se referirán como X0, X1, X2 y X3, donde X es un número binario de 6 bits determinante de la versión particular de circuito que ha sido seleccionada. Cada vía de acceso del circuito puede ser escrito utilizando instrucciones de salida. El contenido de las vías de acceso de entrada/salida puede ser leído utilizando instrucciones de entrada. Dos de estas vías de acceso están utilizadas como puertos de 8 bits de entrada/salida, mientras que las dos restantes son el temporizador de 8 bits y la vía de acceso de control de interrupciones locales. A las dos primeras se les asigna la dirección binaria X00, para la vía de acceso A; X01, para la vía de acceso B; X10, para el control local de interrupciones; X11, para el temporizador.

El temporizador de este circuito se utiliza para establecer retardos por programa. Bajo el punto de vista de programa, el temporizador es un registro de 8 bits, direccionable por medio de las instrucciones de salida del microprocesador F8. Los códigos de temporizado, calculados por programa, son cargados en el acumulador y posteriormente transferidos al temporizador. Para realizar esta función se realiza una instrucción de salida. Una vez que ha sido cargado el temporizador inicia su contaje en sentido descendente. Pueden ser seleccionados retardos entre 0 y 254 contajes. El temporizador decrementa un paso cada 31 ciclos de reloj «Φ». Por tanto el contador puede llegar a contar hasta 7905 ciclos de reloj (para un sistema con reloj de 2 MHz, un ciclo de contaje ocurre cada 500 ns). Duraciones mayores pueden conseguirse mediante el contaje de múltiples interrupciones temporizadas. Si el temporizador se carga con todo «unos» para su contaje.

Este circuito dispone asimismo de dos tipos distintos de interrupciones: interna o externa. La interrupción interna puede ser generada por el temporizador programable, mientras que la interrupción externa está generada por lógica externa al sistema. Un circuito de control local de interrupciones formado por dos biestables o flip-flops es el corazón del sistema; estos biestables son el bit de selección y el bit de autorización de interrupción.

4.3.18 Entrada/salida universal (10696)

El circuito de entrada/salida universal 10696 está desarrollado por ROCKWELL, para expandir la posibilidad de entrada/salidas de su microprocesador PPS. Según se puede observar en la figura 4.30 está compuesto por varios bloques entre los que se destacan: zona de entradas, formada por 12 bits, agrupados en 3 grupos de 4, seleccionables; zona de amplificadores de salida, formada asimismo por 12 bits en 3 grupos de 4; lógica de selección de los anteriores; decodificador de instrucciones; circuito lógico de selección de la unidad, y un circuito amplificador del reloj.

Para programar este circuito es preciso emitir 2 palabras de control desde la unidad central. La primera palabra es la señal de conexión de entrada/salida. La segunda palabra contiene la dirección del circuito periférico seleccionado y la instrucción que se le solicita. En esta segunda palabra, 4 bits indican la dirección del periférico, dirección que será comparada con las 4 entradas de preselección de que dispone, y los otros 4 bits especifican cuál de los 3 grupos de entrada o de los 3 grupos de salida va a ser activo.

Este circuito es de manejo simple y como puede observarse dispone de una decodificación propia de su dirección, de tal modo que sin necesidad de circuito externo pueden

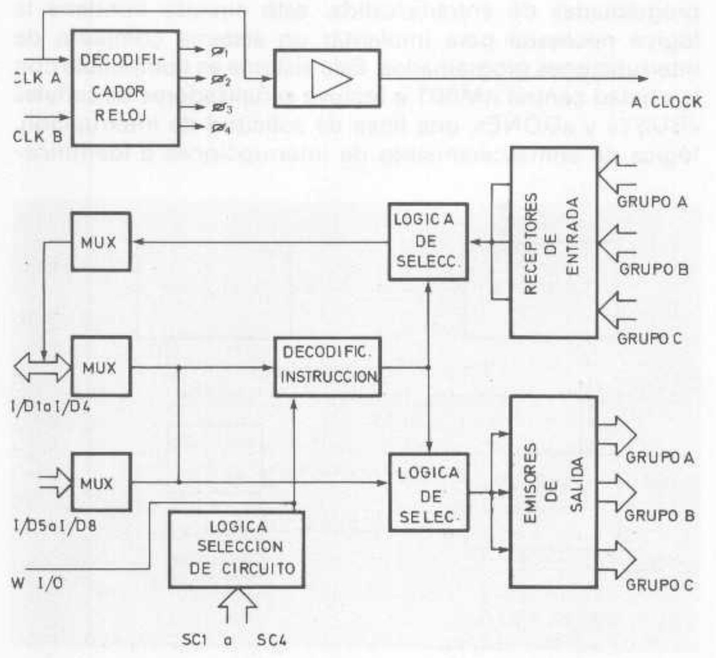


Figura 4.30 Entrada/salida universal 10696

conectarse hasta 16 unidades distintas al mismo microprocesador, lo cual permite obtener hasta 192 bits de entrada y otros tantos 192 bits de salida.

4.3.19 Controlador de entrada/salida (mN603)

El circuito mN603 es un controlador de entrada/salida desarrollado por Data General para conexión a su microprocesador MICRONOVA. Provee una total capacidad funcional respecto del bus de entrada/salida NOVA de 47 vías, aún cuando eléctricamente tiene una implantación más simple realizada en 16 bits [6]. El circuito decodifica un tren de datos codificados desde la unidad central y presenta una interfaz bidireccional paralela de 16 bits, 4 bits de funciones codificadas, y una señal de validación de función hacia el periférico, permitiendo de este modo una interfaz de simplicidad razonable (fig. 4.31).

Asimismo este circuito incluye una identificación específica de circuito, lógica de interrupción «BUSY/DONE», y una capacidad de interrupción enmascarada por periférico. Con destino a controladores orientados a bloques de datos, incluye una serie de señales de validación interactiva del bus de datos de canal y contadores de la totalidad de bits de dirección y de palabra.

Para construir un controlador completo de entrada/salida es preciso añadir algunos circuitos auxiliares, tales como un generador de reloj mN640, un transceptor de entrada/salida mN636, un circuito de prioridad de canal de datos e interrupciones, una lógica de decodificación formada por 4 bits, y puertas lógicas adicionales para realizar interfaz con el periférico, así como amplificación de señales.

Cada vez que la unidad central extrae una instrucción de entrada/salida desde la memoria, transmite la instrucción completa sobre el bus de entrada/salida. Cada controlador de entrada/salida conectado al bus recibe una copia de esta instrucción e internamente la decodifica. Si el circuito determina que la instrucción va dirigida hacia él, la ejecuta. Dado que este circuito dispone de una copia exacta de la instrucción de entrada/salida puede emular todas las señales de control programadas de entrada/salida utilizadas en el bus NOVA/ECLIPSE.

Adicionalmente a la decodificación de instrucciones programadas de entrada/salida, este circuito contiene la lógica necesaria para implantar un sistema completo de interrupciones programadas. Este sistema es compatible con la unidad central nM601 e incluye señalizadores de señales «BUSY» y «DONE», una línea de solicitud de interrupción, lógica de enmascaramiento de interrupciones e identificación de fuentes de interrupciones. Asimismo está provisto de la lógica necesaria para realizar las transferencias de datos por canal. Esto incluye la lógica de solicitud y dos registros internos, un registro de dirección de memoria y un registro de contaje de palabras, que controlan la transferencia de datos.

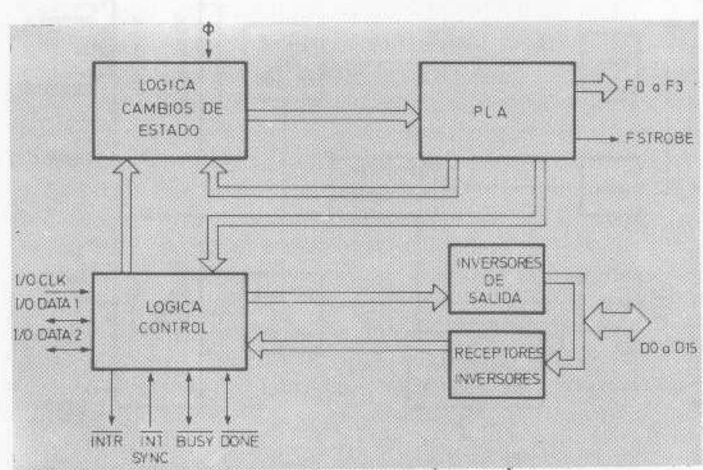


Figura 4.31 Controlador de entrada/salida mN603

Este controlador dirige automáticamente todos los protocolos basados en el bus de entrada/salida MICRO NOVA. Estos protocolos incluyen transferencia de datos, instrucciones programadas de entrada/salida, y todas las señales de sincronización necesarias para interrupción y proceso de datos. Está compuesto por 4 bloques funcionales fundamentales: la vía de acceso (port) de datos periféricos que provee al circuito con un bus de datos bidireccional de 16 bits y 4 líneas de control de funciones codificadas; la vía de acceso de datos de entrada/salida que permite conversiones serie paralelo de la información transferida hacia o desde el bus de entrada/salida; los registros internos, vías de datos, y organización lógica decodifican las instrucciones y conectan las dos vías de acceso de datos; el circuito de solicitud de control integra las solicitudes de servicio en el resto del sistema MICRO NOVA.

Estas cuatro áreas principales son:

«Port» de datos periférico. Este circuito procesa la transferencia de datos entre una unidad periférica y este controlador. El «port» de datos periférico incluye las 16 líneas de datos (D0 a D15), las 4 líneas codificadas de función (F0 a F3), y el reloj (FSTROBE). Las 16 líneas de datos transfieren éstos, hacia y desde la unidad periférica. Los códigos de función, con su reloj de sincronización, controlan el acceso de datos hacia estas líneas.

«Port» de datos de entrada/salida. Esta sección controla la transferencia entre el bus de entrada/salida y el propio controlador. La vía de acceso de datos de entrada/salida incluye 2 líneas de datos serie (I/O DATA 1 y I/O DATA 2), su señal de validación (I/O CLOCK), y una línea de control de transmisión/recepción (I/O INPUNT). Las líneas I/O DATA 1 y I/O DATA 2 reciben instrucciones y datos desde el bus de entrada/salida y transmiten datos hacia el propio bus de comunicación. La señal I/O CLOCK valida la señal transmitida en estas líneas de datos. I/O INPUT indica cuando está transmitiendo o recibiendo el controlador por las líneas de datos.

Estructura interna. Un conjunto de registros internos, junto con las vías de datos que los interconectan, relacionan el «port» de datos de entrada/salida con el «port» de datos periférico. Todo el control de transferencias de información y solicitudes de servicio proviene desde el conjunto lógico interno y la lógica de cambio de estado. Dos de estos registros internos son utilizados durante secuencias de ruptura de datos y pueden ser reemplazados por registros externos. Las solicitudes de interrupción y las solicitudes de ruptura de datos son controlados por la lógica de solicitud.

Control de solicitudes. Este bloque trata las interrupciones de programa y las posibilidades de ruptura de datos. Cada una de estas posibilidades tienen redes independientes de solicitud y prioridad. La mayor parte de esta lógica es interna al propio circuito. No obstante el diseñador puede elegir utilizar tanto lógica interna como externa para las señales «BUSY» y «DONE».

4.3.20 Controlador de bus de entrada/salida (9442)

El 9442 es un circuito de control de interfaz paralelo de entrada/salida diseñado por FAIRCHILD como periférico de su microprocesador 9440 «micro Flame». La figura 4.32

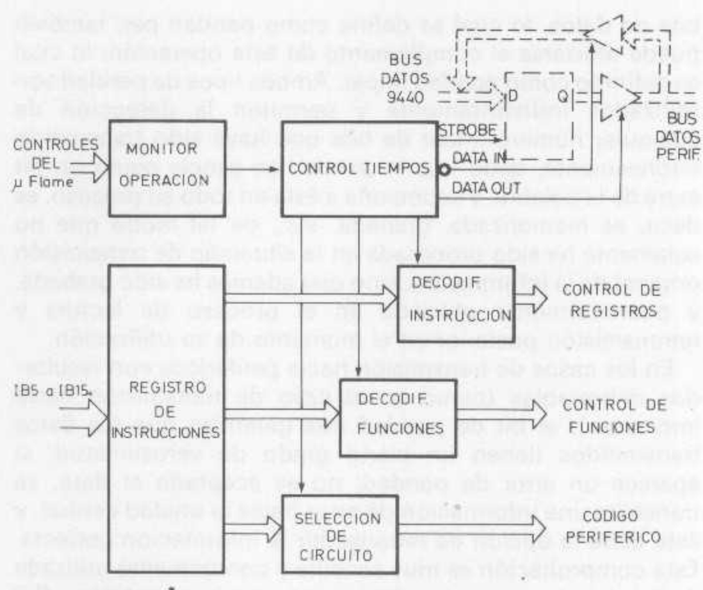


Figura 4.32 Controlador de bus de entrada/salida 9442

indica un bloque funcional de su estructura interna. Está realizado partiendo de tecnología bipolar (I^3L), y permite compatibilidad total TTL para todas las vías de entrada/salida.

Fundamentalmente este circuito almacena y decodifica las instrucciones de entrada/salida. Permite una expansión del bus de control de entrada/salida y simultáneamente genera las señales de sincronización precisas. Permite una utilización centralizada junto al microprocesador para expandir el bus de entrada/salida, o bien ser situado localmente junto

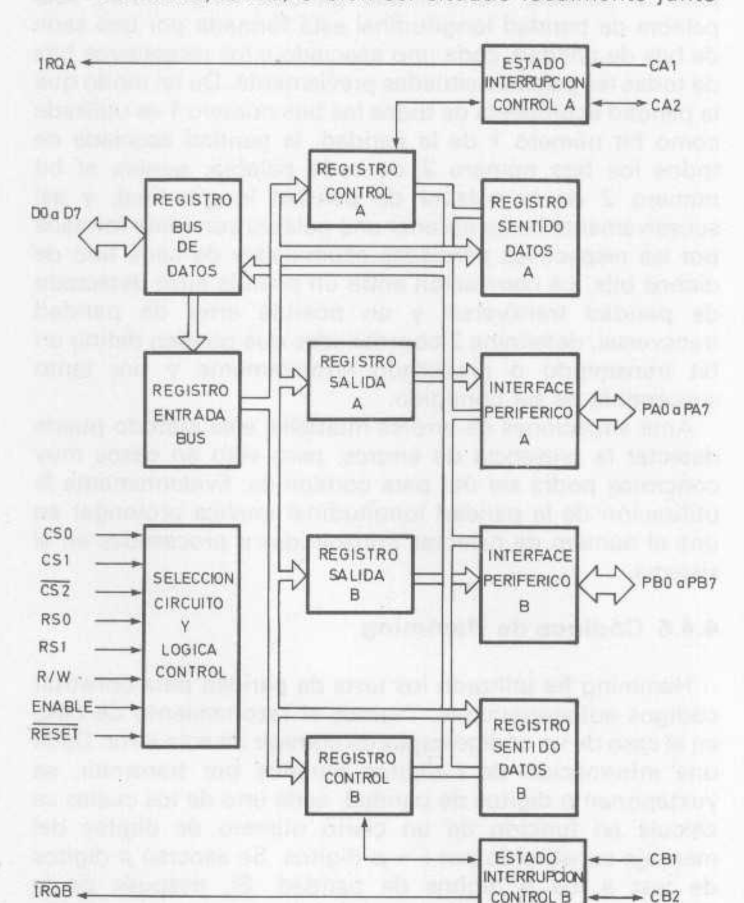


Figura 4.33 Adaptador de interfaz periférica 6821

cada periférico con el fin de efectuar las funciones de control a él asociadas. Este circuito genera todas las señales de control precisas pero no procesa por sí mismo el bus de datos. Para ello es preciso añadir una lógica externa formada por transceptores y registros para memorización y sincronización transitoria de la información.

4.3.21 Adaptador de interfaz periférica (6821)

El 6821 es un circuito desarrollado por MOTOROLA como adaptador de interfaz periférica para su microprocesador 6800. Permite la interconexión de la unidad central a los periféricos a través de 2 buses bidireccionales de 8 bits y 4 líneas de control. Generalmente no se precisa otra lógica externa complementaria (fig. 4.33).

La configuración funcional de este circuito es programada por el microprocesador durante el período de inicialización del sistema. Cada una de las líneas de datos periféricas puede ser programada para actuar como entrada o como salida, y cada una de las 4 líneas de control/interrupción puede ser programada para una o varias modalidades de control.

Este circuito realiza la interconexión con el 6800 a través de un bus de datos bidireccional de 8 bits, 3 líneas de selección de circuito específico, 2 líneas de selección de registros, 2 líneas de solicitud de interrupción, la línea de lectura/escritura y las líneas de validación y restauración.

4.4 PROTECCIÓN DE LA INFORMACIÓN

4.4.1 Fiabilidad de los datos

En cualquier proceso de transmisión de señales eléctricas que contienen información válida, varias causas pueden provocar interferencias en la correcta transmisión, provocando por tanto la aparición de códigos o datos erróneos que pueden falsear los resultados pretendidos. En estos casos se acude a la utilización de códigos, generalmente formados añadiendo algún tipo de información a las propias palabras de datos transmitidas, que permitan, o bien detectar los errores (códigos autodetectores), o bien corregir estos errores total o parcialmente (códigos autocorrectores). A menudo a este tipo de codificación se les denomina códigos redundantes, es decir que es preciso utilizar un número de bits superior al estrictamente necesario para la codificación de la información. Entre los motivos que pueden provocar estas alteraciones en la fidelidad de la información transmitida existen muchos, tales como interferencia entre señales vecinas (por causa de acoplamientos inductivos o capacitivos), interferencias externas causadas por señales eléctricas ajenas al sistema (tales como líneas de transmisión de potencia), cortes momentáneos en la transmisión (debidos a fallos de contactos, fallos instantáneos de la alimentación), y cualquier otro motivo que pudiera distorsionar la codificación real transmitida.

4.4.2 Echoplex

El sistema más simple utilizado para garantizar la fiabilidad de los datos transmitidos es el Echoplex. Este método consiste en lo siguiente: cuando la unidad central transmite una cierta palabra binaria, el periférico la recibe, la procesa, y la retransmite hacia la unidad central; posteriormente el programa de la unidad central verifica la identidad entre la palabra recibida como respuesta y la palabra original que se

transmitió; si existe total correspondencia sigue adelante el programa y por tanto la secuencia de transmisión; si existe divergencia entre el significado de la palabra transmitida y la recibida, se inicia una rutina de corrección habiendo detectado por tanto un error de transmisión en la línea.

Este método asegura una fiabilidad muy elevada en los datos transmitidos, pero como contrapartida introduce posibles desventajas. Por un lado, si la transmisión es por una sola vía de datos, provoca un ralentizamiento de la velocidad de transmisión, por cuanto que a cada palabra transmitida es preciso efectuar una conmutación de inversión de sentido de transmisión y la retransmisión del eco, lo cual aumenta notablemente los tiempos muertos en la transmisión; por otra parte, para mejorar esta velocidad de transmisión puede acudirse a la utilización de dobles vías de datos pero ello implica costos adicionales notablemente altos por cuanto es preciso disponer de dos canales simultáneos, uno para transmisión de la información desde la unidad central hasta el periférico y otro para la retransmisión posterior con fines de validación.

4.4.3 Datos no regenerables

El método Echoplex es válido cuando el proceso de la información es efectuado en tiempo real, es decir se procesa la información simultáneamente a su generación, de tal modo que si se detecta un error en la transmisión, dado que la información todavía está presente, es posible retransmitirla. A menudo se presentan circunstancias en las cuales no es posible efectuar la retransmisión, fundamentalmente cuando el origen de los datos procede de una información diferida, tal como una grabación en disco magnético, en cinta, etc.

En cualquiera de estos casos se deben considerar las dos transmisiones que lleva implícita la memorización de datos; la primera transmisión se efectúa cuando la unidad central emitió la transmisión hacia el periférico, en aquel momento pudo ser corregido cualquier error por medio del Echoplex; la segunda transmisión asociada es la que se efectúa cuando se lee esta información en un momento posterior, momento en el cual la única fuente de información es la grabación existente en disco o cinta; si en este momento aparece un error, fuese por lectura del soporte magnético o bien por defecto de transmisión, puede alcanzarse fácilmente una situación de datos irrecuperables. Cuando deban efectuarse transmisiones de este tipo es preciso utilizar algún tipo de codificación que nos garantice que la información realmente ha sido la correcta.

Por otra parte, si se ha producido un defecto de grabación en el soporte magnético, el método Echoplex no nos garantiza la fiabilidad de los datos originalmente grabados, puesto que solamente permite certificar la calidad de la transmisión. Otra situación equivalente es la que se presenta cuando la transmisión va destinada a periféricos en los cuales la información es registrada de modo imborrable, por ejemplo en una impresora; en este caso, si se presenta un defecto en la transmisión, el código erróneo ya ha sido impreso; aunque puede regenerarse o bien retransmitirse esta información, ya ha aparecido el error sobre el papel.

4.4.4 Paridad

En las situaciones descritas en el apartado anterior puede mejorarse sensiblemente la correcta fiabilidad de los datos mediante la utilización de un bit auxiliar añadido a la palabra de datos; este bit se define como paridad y consiste en el resultado de efectuar la función O-exclusiva entre todos los bits de datos, lo cual se define como paridad par; también puede utilizarse el complemento de esta operación, lo cual es definido como paridad impar. Ambos tipos de paridad son utilizados indistintamente y permiten la detección de cualquier número impar de bits que haya sido transmitido erróneamente, dado que la paridad se asocia como un bit extra de la palabra, y acompaña a ésta en todo su proceso, es decir, es memorizada, grabada, etc., de tal modo que no solamente ha sido procesada en la situación de transmisión original de la información, sino que además ha sido grabada, y posteriormente utilizada en el proceso de lectura y retransmisión posterior en el momento de su utilización.

En los casos de transmisión hacia periféricos con resultados imborrables (como en el caso de transmisión hacia impresora), el bit de paridad nos garantiza que los datos transmitidos tienen un cierto grado de verosimilitud; si aparece un error de paridad, no es aceptado el dato, se transmite una información de error hacia la unidad central, y ésta tiene la opción de retransmitir la información correcta. Esta comprobación es muy sencilla y comúnmente utilizada dado el bajo costo que representa su implantación. Sin embargo, un simple bit de paridad puede fallar fácilmente ya que no puede detectar un número par de errores en bits.

4.4.5 Paridad longitudinal

La paridad anteriormente definida es denominada comúnmente paridad transversal. Este nombre proviene de que en equipos tales como perforadores de cinta de papel o bien grabadores de cinta magnética, el bit de paridad antes indicado se asocia transversalmente al sentido de avance de la cinta. A menudo se utiliza también una palabra completa para la paridad, denominada paridad longitudinal; esta palabra de paridad longitudinal está formada por una serie de bits de paridad, cada uno asociado a los respectivos bits de todas las palabras situadas previamente. De tal modo que la paridad acumulada de todos los bits número 1 es utilizada como bit número 1 de la paridad; la paridad asociada de todos los bits número 2 de cada palabra, genera el bit número 2 de la palabra de paridad longitudinal, y así sucesivamente hasta obtener una palabra completa formada por las respectivas paridades acumuladas de cada uno de dichos bits. La correlación entre un posible error detectado de paridad transversal y un posible error de paridad transversal, determina 2 coordenadas que pueden definir un bit transmitido o procesado erróneamente y por tanto susceptible de ser corregido.

Ante situaciones de errores múltiples este método puede detectar la presencia de errores, pero sólo en casos muy concretos podrá ser útil para corregirlos. Evidentemente la utilización de la paridad longitudinal implica prolongar en una el número de palabras transmitidas o procesadas en el sistema.

4.4.6 Códigos de Hamming

Hamming ha utilizado los tests de paridad para construir códigos autocorrectores. Veamos el razonamiento de base en el caso de un código capaz de corregir un solo error. Dada una información de i digitos binarios por transmitir, se yuxtaponen p digitos de paridad, cada uno de los cuales se calcula en función de un cierto número de digitos del mensaje constituido por $i + p$ digitos. Se asocian p digitos de test a los p digitos de paridad. Si, después de la transmisión, la paridad no se satisface para un determinado digito de paridad, se conviene en dar el valor 1 al digito de

test correspondiente; en caso contrario se le atribuye el valor 0. Se escoge *p* lo suficientemente grande para que los digitos de test puedan codificar, o bien la ausencia de error (los *p* bits con 0), o bien la posición del error en el mensaje de $i + p$ digitos, suponiendo que haya solamente uno. Una vez obtenida la posición del error, basta invertir el valor de dicho bit para obtener el mensaje correcto. La condición necesaria para que *p digitos de test, puedan codificar los* $i + p$ bits del mensaje, detectando y corrigiendo un error, implica asociar a *p* un valor tal que verifique $2^p \geqslant i + p + 1$.

Los códigos autocorrectores de Hamming tienen el inconveniente de exigir la generación, proceso y transmisión de un número suplementario de bits, lo cual complica y encarece el sistema completo de comunicación. Como orientación se puede indicar que para garantizar la corrección de un bit en error, es preciso utilizar: para la transmisión de 1 bit de datos, 2 bits de paridad; para 4 bits de datos, 3 bits de paridad; para 11 bits de datos, 4 bits de paridad; para 26 bits de datos, 5 bits de paridad; para 57 bits de datos, 6 bits de paridad. Puede observarse que utilizando palabras de gran longitud, estos códigos pueden ser eficientes, mientras que para palabras de corta longitud, tal como las comúnmente utilizadas (de 4, 8 o 16 bits), los códigos de Hamming implican un bajo rendimiento en la utilización del sistema.

Si en lugar de pretender corregir un solo error se pretenden corregir mayor número de errores, el número de bits de paridad asociados crece rápidamente llegando a extremos que provocan una muy baja rentabilidad del sistema.

4.4.7 CRC

Las siglas CRC corresponden a las palabras inglesas Cyclic Redundancy Checksum, que podría traducirse por «total acumulado de redundancia cíclica». Este procedimiento hace uso de las propiedades de la división de polinomios.

Eligiendo apropiadamente el polinomio generador es altamente improbable que pueda aparecer una condición de errores que den las características de no detectable. Por tanto la garantía de detección de posibles errores en todo el proceso de datos es, casi, plenamente garantizada.

En la figura 4.34 puede verse un diagrama de bloques de una posible generación de CRC paralelo; nótese que los circuitos para generación de CRC en serie son distintos aún cuando basados en las mismas propiedades [7].

Según el diagrama de bloques, los pasos que se siguen para obtener el CRC de una serie de palabras transmitidas son los siguientes:

Primero. Cada carácter de datos es sumado al contenido del registro CRC sin arrastres, por tanto se realiza la función O-exclusiva entre cada bit y el correspondiente bit del registro CRC.

Segundo. Esta información sufre seguidamente un decalaje circular a derechas de un lugar, de tal modo que cada bit es copiado en la posición adyacente del registro CRC.

Tercero. Si el bit entrado en la posición P del CRC es «1», los bits entrados en las posiciones 2, 3, 4 y 5 son invertidos.

Cuarto. Se repiten los pasos desde 1 hasta el 3 por cada carácter de datos que se transmita.

Quinto. El contenido de todas las posiciones del registro CRC, excepto las posiciones 2 y 4 son invertidos y el carácter resultante es transmitido seguidamente a la información.

Obsérvese que este circuito requiere un impulso de reloj

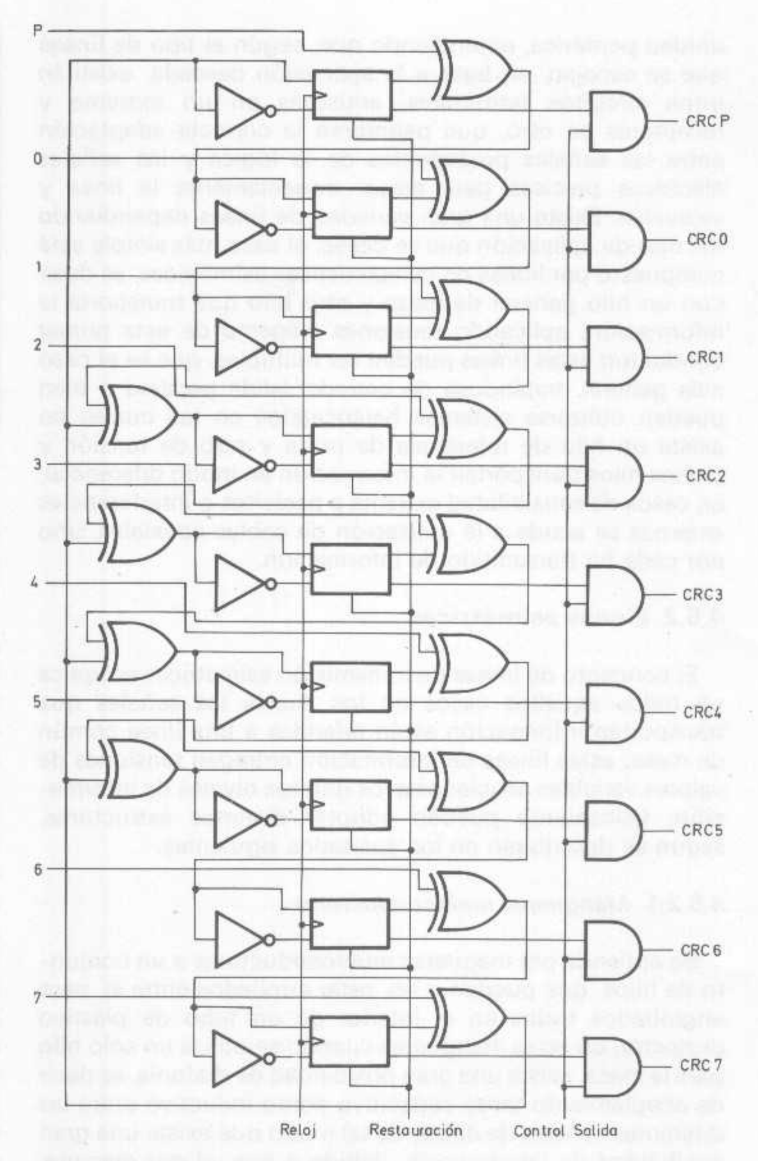


Figura 4.34 Generador de CRC paralelo

por cada carácter de datos, y que no es preciso efectuar ningún decalaje suplementario después del último carácter de datos.

Los circuitos generadores y detectores de CRC son ciertamente más complejos que los equivalentes para una simple generación o detección de paridad, pero el grado de seguridad que comportan y el hecho de no requerir vías extras en los buses de transmisión, ni incremento notable del tiempo debido a transmisión de muchos caracteres suplementarios (solamente un carácter al final de un bloque de datos), lo hacen muy aconsejable en todos aquellos casos en los que se pretenda una elevada fiabilidad sin existir posibilidad de regeneración de los datos iniciales. Por ejemplo cuando se pretenda grabar o leer desde discos o cintas magnéticas o cualquier otro tipo de soporte similar con lectura diferida.

4.5 LÍNEAS DE TRANSMISIÓN

4.5.1 Tipos de líneas de transmisión

Se denominan líneas de transmisión al conjunto de hilos que realizan el enlace físico entre la unidad central y la

unidad periférica, entendiendo que, según el tipo de líneas que se escojan, en base a la aplicación deseada, existirán unos circuitos terminales, emisores en un extremo y receptores en otro, que permitirán la correcta adaptación entre las señales procedentes de la lógica y las señales eléctricas precisas para atacar correctamente la línea y viceversa. Existe una gran variedad de líneas dependiendo del tipo de aplicación que se desee; el caso más simple está compuesto por líneas de características asimétricas, es decir con un hilo general de masa y otro hilo que transporta la información, aplicando tensiones respecto de este primer conductor; estas líneas pueden ser múltiples, que es el caso más general, tratándose de entrada/salida paralelo o bien pueden utilizarse sistemas balanceados en los cuales no existe un hilo de referencia de masa y otro de tensión y ambos hilos transportan la información en modo diferencial; en casos de sensibilidad extrema a parásitos o interferencias externas se acude a la utilización de cables coaxiales, uno por cada bit transmitido de información.

4.5.2 Líneas asimétricas

El concepto de líneas de transmisión asimétricas se aplica en todos aquellos casos en los cuales las señales que transportan información están referidas a una línea común de masa; estas líneas de información entregan tensiones de valores variables asociados a los ditintos niveles de información; físicamente pueden adoptar distintas estructuras, según se describirán en los apartados siguientes.

4.5.2.1 *Mangueras multiconductores*

Se entiende por magueras multiconductores a un conjunto de hilos, que pueden o no, estar arrollados entre sí, pero englobados todos en el interior de un tubo de plástico protector. En estas mangueras cuando se utiliza un solo hilo para la masa, existe una gran posibilidad de diafonía, es decir de acoplamiento tanto capacitivo como inductivo entre las diferentes señales de datos, de tal modo que existe una gran posibilidad de interferencia, debido a que, si por ejemplo, estamos trabajando con informaciones de 8 bits, y 7 de ellos conmutan simultáneamente, el octavo sufrirá una fuerte influencia, de tal modo que presentará un impulso como señal diferencial de las conmutaciones del resto de hilos. Las líneas de este tipo son bastante económicas pero no pueden ser utilizadas para grandes distancias debido al efecto antes enunciado.

4.5.2.2 *Pares trenzados*

La conexión por pares trenzados («Twisted Pair»), es un método bastante económico y mucho más fiable que el de las mangueras multiconductores; este método consiste en que a cada señal se le asocia su propia masa de referencia, de tal modo que ambos conductores van arrollando entre sí dando una vuelta completa entre si cada 2 a 6 cm de longitud. El hecho de este arrollamiento provoca un efecto similar a un blindaje, de tal modo que las corrientes de ida por el conductor de señal son canceladas por la corriente de retorno por el conductor de masa, minimizando el campo externo generado y por tanto reduciendo las posibles interferencias a cables vecinos. Por otra parte este efecto de blindaje generado por la masa arrollada inmuniza bastante el hilo de señal respecto a interferencias externas, puesto que cualquier señal acoplada se induce simultáneamente en ambos conductores, de modo tal que la tensión diferencial

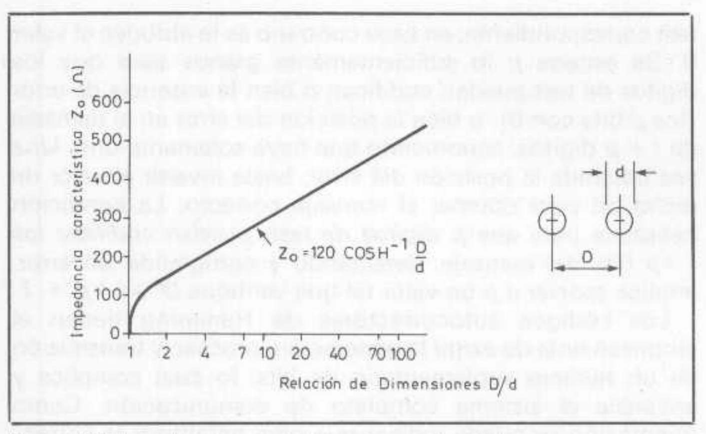


Figura 4.35 Impedancia característica de pares trenzados

aparecida es mínima. La posible tensión en modo común es anulada por el hecho de la conexión a cero voltios del conductor de retorno.

Las líneas de pares trenzados presentan una impedancia característica comprendida normalmente entre 100 y 200 ohmios [8], según se puede observar en el gráfico (fig. 4.35). En este gráfico se indica la impedancia característica de una línea en función de las relaciones entre el diámetro de los conductores y la distancia entre centros.

De todos modos la impedancia característica de cualquier línea desconocida puede ser determinada utilizando un medidor RX, de tal modo que la impedancia Z de la línea se calculará a partir de la impedancia medida de cortocircuito Z_{SC} y la impedancia medida en circuito abierto Z_{OC}, de tal modo que $Z = (Z_{SC} \cdot Z_{OC})^{1/2}$. Normalmente la impedancia de la línea es considerablemente inferior a las impedancias de salida del emisor o de entrada del receptor. Esto es deseable puesto que cuando se conecten múltiples emisores o receptores a una línea no se afectará su impedancia, con lo que se impiden situaciones críticas de acoplamiento. Con el fin de prevenir reflexiones eléctricas en la línea, es conveniente intercalar una red terminal, cuya impedancia sea equivalente a la característica de la línea. Asimismo esto reduce la impedancia general del conjunto y por tanto aumenta la inmunidad contra inducciones externas.

4.5.2.3 *Cable plano*

El cable plano («Flat Cable») es otro tipo de línea de conducción desarrollado inicialmente por 3M. Consiste en una serie de conductores paralelos formando una cinta plana, toda ella aislada por una envoltura de PVC. El tipo más comúnmente utilizado es el 3365 (fig. 4.36), que presenta un diámetro de cada uno de los conductores de 0,32 mm, lo cual implica una sección de 0,08 mm^2. Existe en diferentes anchos hasta un máximo de 50 hilos. Su impedancia característica es de 100 ohmios. La velocidad de propagación de la señal eléctrica medida es de 4,6 ns por metro lineal. Dada la pequeña distancia entre conductores, presenta unas características muy similares a las de los pares trenzados, permitiendo un montaje mucho más limpio y cómodo, puesto que los conectores diseñados para este tipo

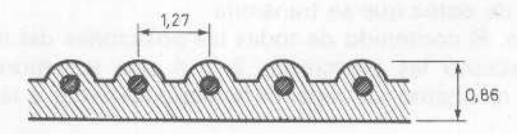


Figura 4.36 Sección del cable plano 3365

de cable trabajan por un sistema de prensado, es decir el cable plano se sitúa sobre el conector, se prensa y los terminales del conector ya están diseñados para pinzar individualmente cada uno de los conductores atravesando la envoltura protectora. Cuando se deseen características similares a las de los pares trenzados, usualmente se van destinando conexiones contiguas alternativamente a señal y masa, de tal modo que entre cada dos señales existe una masa de blindaje intermedio. Asimismo es de aplicación el comentario que se ha hecho sobre los pares trenzados respecto a la red terminal de resistencias para equilibrio de impedancias y por tanto supresión de reflexiones eléctricas en la línea.

4.5.2.4 *Cable coaxial*

El cable coaxial no es excesivamente utilizado para transmisiones de múltiples bits en paralelo por cuanto exige un conductor individual por cada bit, haciéndose por tanto sumamente engorroso el proceso de construcción e instalación de mangueras. No obstante, en determinadas ocasiones puede llegar a ser imprescindible. La impedancia de los cables coaxiales usualmente es bastante inferior a la de los pares trenzados o los cables planos y está normalmente comprendida entre valores de 50 a 75 ohmios [8]. El gráfico (fig. 4.37) indica la impedancia característica de estos cables coaxiales en función de los diámetros del conductor interno y el diámetro interior del conductor externo. Este tipo de conexión es el más fiable en cuanto a inmunidad a parásitos o ruidos, pero presenta el inconveniente de su elevada capacidad intrínseca, por lo que presenta con gran facilidad atenuación de las componentes de frecuencias más elevadas.

4.5.3 Líneas balanceadas

Un tipo distinto de transmisión es la que se efectúa por líneas balanceadas y en este caso no existe un conductor destinado a la masa y otro a la señal sino que ambos conductores transmiten la señal en modo diferencial. Evidentemente, tanto los emisores como los receptores deben estar diseñados a este efecto, lo cual implica la utilización de circuitos especiales terminales. Dado que la transmisión se efectúa en modo diferencial, las posibles inducciones externas sobre la línea, que generalmente son de modo común, no provocan alteración alguna en la calidad de la señal transmitida. Las propias características de este tipo de enlace exigen la utilización de conductores asimismo balanceados; por tanto pueden utilizarse cables planos o trenzados, pero no es recomendable la utilización de cables coaxiales por su asimetría intrínseca. En la figura se ilustra una aplicación típica de transmisión en cable balanceado entre dos equipos de procesos de datos. Las resistencias terminales R_T deben adoptar un valor igual a la mitad de la impedancia característica del cable de transmisión (fig. 4.38).

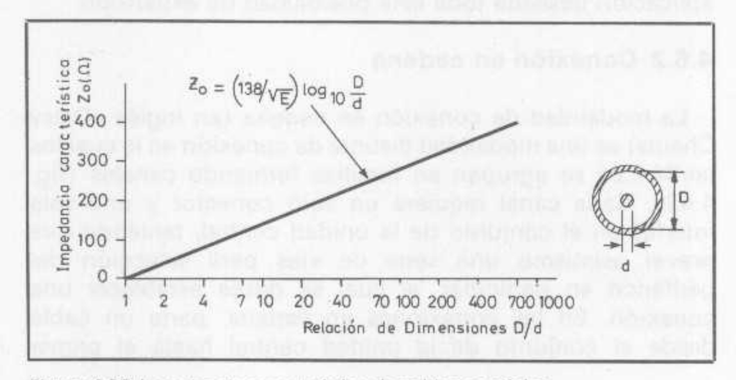


Figura 4.37 Impedancia característica de cables coaxiales

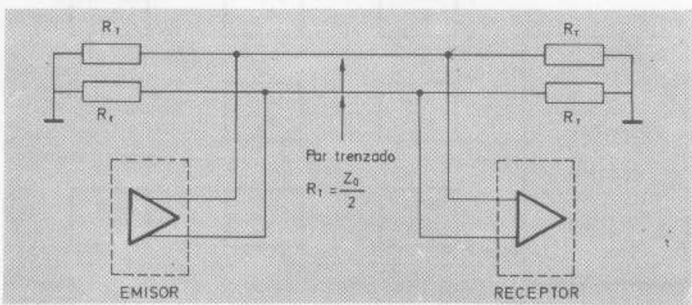


Figura 4.38 Terminación de líneas balanceadas

4.5.4 Aislamiento galvánico

En determinados equipos puede ser imperativo efectuar un aislamiento total entre las tensiones eléctricas respectivas; por tanto, para la transmisión de la información es preciso adoptar otras técnicas, tales como las seguidamente descritas.

La aplicación más primitiva fue la utilizada para conexión de teletipos, en las cuales el aislamiento entre el ordenador central y el circuito propio del teletipo se realizaba por medio de un relé, pero en los modernos métodos de transmisión en que las velocidades son bastante elevadas, cualquier método electromecánico (tal como el propio relé indicado) adolece de una inercia que lo hace absolutamente inadecuado para estas aplicaciones.

4.5.4.1 *Acopladores optoelectrónicos*

El método más comúnmente utilizado consiste en emplear acopladores optoelectrónicos, consistentes en un conjunto formado por un diodo electroluminiscente (LED) y un fototransistor o fotodiodo, todo ello incluido en una única cápsula (fig. 4.39). La señal a transmitir se aplica al diodo luminiscente, el cual emite radiación infrarroja que incide sobre el fotorreceptor cuya resistencia interna varía en consonancia. Entre el diodo emisor y el elemento receptor existe un aislamiento que garantiza inmunidad contra tensiones del orden de kilovoltios, garantizando por tanto una absoluta independencia de las tensiones respectivas. Comúnmente se sitúa el acoplador optoelectrónico muy próximo al circuito receptor, puesto que de este modo la señal transmitida por el cable de conexión es la corriente de

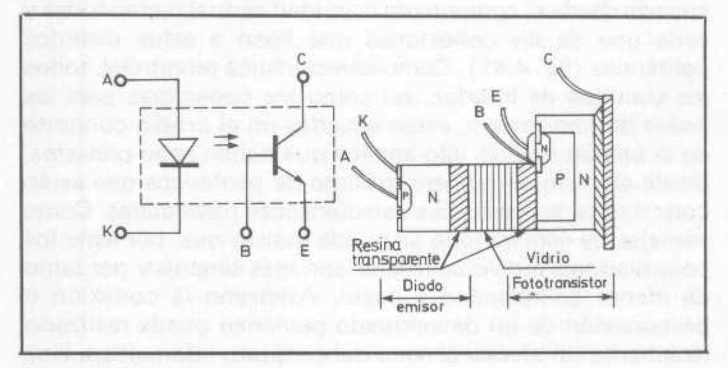


Figura 4.39 Acoplador optoelectrónico

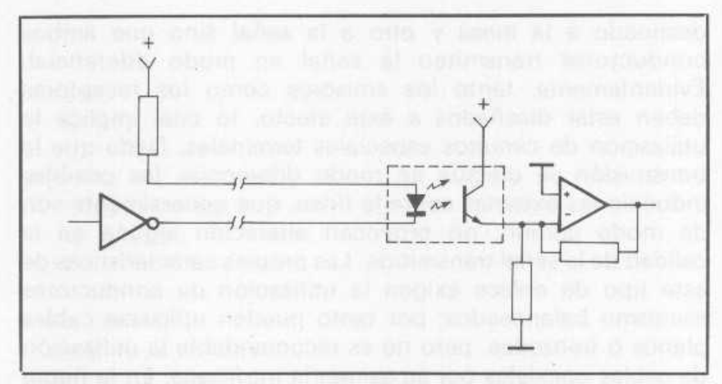


Figura 4.40 Conexión por medio de acopladores optoelectrónicos

baja impedancia que ataca al diodo emisor, mientras que la señal que aparece en bornas del circuito receptor es de muy bajo nivel y podría haber sido fácilmente interferida (fig. 4.40). Hoy en día existen acopladores optoelectrónicos que permiten transmitir informaciones de frecuencias correspondientes a varios megahertzios.

4.5.4.2 *Fibras ópticas*

Uno de los últimos métodos desarrollados para transmisiones de muy elevada velocidad y absoluto aislamiento galvánico entre equipos es el formado por cables de fibras conductoras ópticas. En este método no existe un cable conductor eléctrico entre ambos circuitos, sino que el circuito que trabaja como emisor dispone de una serie de diodos electroluminiscentes, trabajando normalmente en el campo de la radiación infrarroja, acoplado al extremo de estas fibras. La fibra, con una envoltura opaca protectora, cubre la distancia entre el equipo emisor y el receptor. En el extremo receptor, la otra extremidad del cable óptico está enfrentado a un elemento captador optoelectrónico y convierte estas radiaciones luminosas en señales eléctricas aptas para excitar los circuitos periféricos asociados. Los enlaces por medio de fibras ópticas presentan una total y completa inmunidad a interferencias eléctricas puesto que no existe señal eléctrica en el propio conductor, por tanto ni reciben ni radian ningún tipo de perturbaciones. No obstante, su elevado costo lo hace impracticable para aplicaciones de uso general y solamente es utilizado en casos muy específicos.

4.6 MODALIDADES DE CONEXIÓN

4.6.1 Conexión radial

La conexión de periféricos puede efectuarse radialmente cuando desde el conjunto de la unidad central parten todas y cada una de las conexiones que ligan a estos distintos periféricos (fig. 4.41). Como característica primordial, todos los circuitos de interfaz, así como los conectores para las líneas de transmisión, están situados en el propio conjunto de la unidad central; ello implica que deben estar previstos, desde el inicio, el número máximo de periféricos que serán conectados, así como sus características particulares. Como ventajas de este método se puede indicar que, por estar los controladores individualizados, son más simples y por tanto de menor complejidad y costo. Asimismo la conexión o desconexión de un determinado periférico puede realizarse fácilmente sin afectar al resto del conjunto informático. Una posible avería en cualquier controlador, interfaz, o línea de

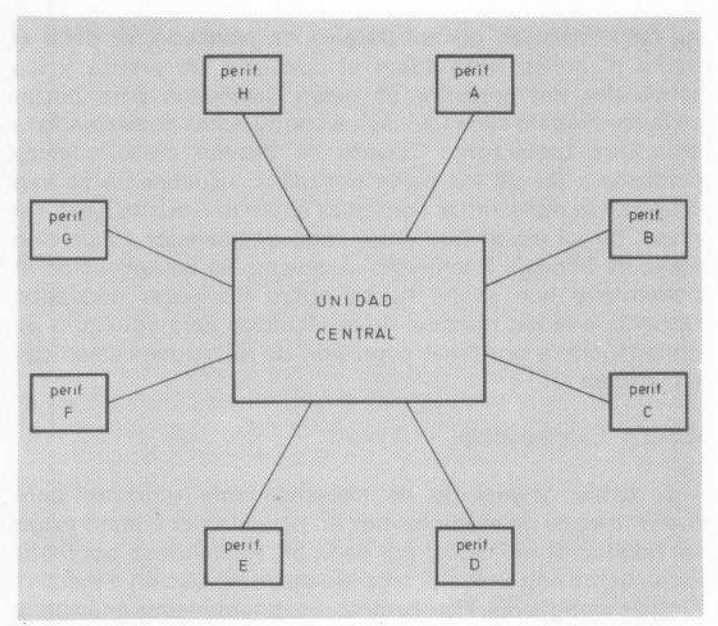


Figura 4.41 Conexión radial de periféricos.

transmisión solamente afecta a su propio periférico y no al resto del equipo.

Como inconvenientes a mencionar se puede indicar que este método de conexión exige una gran longitud de cables, así como una limitación notable en cuanto a expansiones, por cuanto el conjunto de la unidad central debe tener previstas todas las posibilidades en cuanto a conexiones de periféricos, incluyendo sus interfaces y conectores. Esto implica que en sistemas de bajo costo se limita muy fácilmente la posible ampliación de periféricos, por cuanto es reducido el número de conectores e interfaces, no permitiendo por tanto fáciles ampliaciones. Si se pretende solventar este problema añadiendo de origen una amplia diversidad de interfaces y conectores, se corre el riesgo de encarecer inútilmente el sistema por no utilizarse en la aplicación deseada toda esta posibilidad de expansión.

4.6.2 Conexión en cadena

La modalidad de conexión en cadena (en inglés «Daisy Chain») es una modalidad distinta de conexión en la cual los periféricos se agrupan en familias formando canales (fig. 4.42). Cada canal requiere un solo conector y una sola interfaz en el conjunto de la unidad central, teniendo que prever asimismo una serie de vías para selección del periférico en particular, al cual se desea establecer una conexión. En las conexiones en cadena, parte un cable desde el conjunto de la unidad central hasta el primer

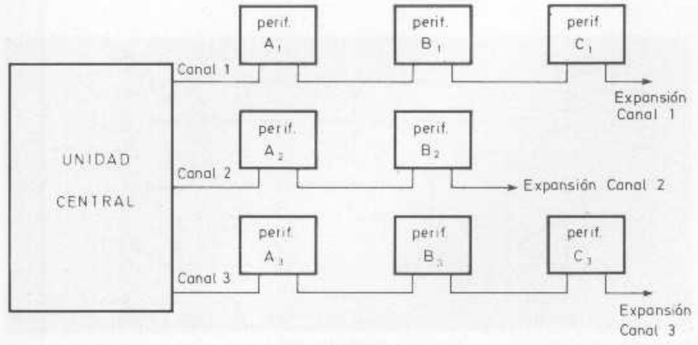


Figura 4.42 Conexión en cadena de periféricos.

periférico de la serie. Otro cable parte desde este periférico y va hasta el segundo, de este otro al tercero y así sucesivamente. La posible ampliación en el número de periféricos se realiza con gran sencillez sin más que añadir cables desde el último periférico de la cadena hasta el próximo que se desee conectar, teniendo presente que normalmente existirá una red de impedancia terminadora situada en el último periférico y que en el caso de ampliación deberá trasladarse hacia el nuevo equipo conectado. Este procedimiento tiene la ventaja de que se pueden realizar con gran sencillez la ampliación de periféricos sin encarecimiento en el resto del equipo, por tanto, puede partirse de equipos de bajo costo y ampliarlos convenientemente. Por otra parte, generalmente se miniminiza la longitud de cables utilizados, simplificando por tanto la instalación.

Como inconveniente debe notarse que una avería, sea en conectores, en líneas de transmisión, o en alguno de los distintos controladores o interfaces, puede provocar que el resto de equipos conectados en este determinado canal sean afectados y por tanto seriamente interferidos. Asimismo en el momento de pretender conectar o desconectar un determinado periférico, si éste es un elemento intermedio de una cadena, debe desconectarse todo el canal para poder efectuar la adición o sustracción.

4.7 CIRCUITOS TERMINALES DE LÍNEA

Por circuitos terminales de línea entendemos tanto los circuitos específicamente transmisores, como los receptores, los conectores y las redes terminales diseñadas en base a los niveles de comunicación que se pretendan emplear.

4.7.1 Niveles normalizados

Una primera característica a tener presente cuando se trate de circuitos terminales de línea son los niveles de tensión que se van a emplear para definir los distintos estados lógicos; existe una gran diversidad de normalizaciones, describiéndose seguidamente las más comúnmente utilizadas.

4.7.1.1 *CCITT V-28 (EIA RS-232-C)*

La recomendación del CCITT V-28 generalmente se utiliza asociada al protocolo recomendado V-24 para transmisiones en serie, pero puede utilizarse asimismo en entrada/salida paralelo en determinadas circunstancias. En esta norma se establece que las señales de control son activas cuando se transmita un nivel positivo por la línea, y los datos se transmiten según la siguiente lógica: al valor binario «0» se le asocia un nivel positivo en línea, mientras que al valor «1» se le asocia un nivel negativo.

Tanto los valores positivos como los negativos deben estar comprendidos en valor absoluto entre 3 y 25 voltios, es decir un «0» lógico debe estar comprendido entre +3 y +25 voltios, mientras que un «1» lógico estará comprendido entre −3 y −25 voltios. La impedancia del circuito receptor no debe ser inferior a 4 kΩ, mientras que la del circuito transmisor debe ser como mínimo de 400 ohmios. Cualquier nivel comprendido entre +3 y −3 voltios se considera como erróneo y es comúnmente utilizado para detectar condiciones de líneas abiertas o cortocircuitadas. Las pendientes de variación de señales eléctricas deben ser como máximo de 30 voltios por microsegundo para impedir situaciones de diafonía entre líneas vecinas.

4.7.1.2 *IBM 360/370*

La interfaz IBM 360/370 establece que las líneas de transmisión presentan una impedancia característica de 50 ohmios, de tal modo que normalmente se trabaja con una salida de 3 voltios (el nivel normalizado es 3,11 voltios) y una corriente típica de 60 miliamperios por línea. Los niveles lógicos establecen que «1» corresponde a la presencia de tensión y corriente en línea, mientras que el «0» corresponde a su ausencia. Estos circuitos deben estar protegidos contra posibles cortocircuitos en la línea, así como establecer un umbral de histéresis en el receptor para impedir conmutaciones erróneas ante señales degradadas. Los circuitos emisores generalmente trabajan con un paso de salida estructurado en forma de seguidor de emisor, permitiendo la utilización de líneas compartidas por distintos emisores y receptores.

4.7.1.3 *TTL*

Los niveles de tensión TTL son quizá los más universalmente utilizados. Están definidos por los niveles equivalentes utilizados en los circuitos integrados del mismo nombre. Se considera un nivel «0» a cualquier valor comprendido entre 0 y 0,8 V, mientras que el nivel «1» es el comprendido entre +2 y +5 V. La carga normal TTL es la equivalente a un consumo de 1,6 miliamperios cuando se aplica un nivel «0»; sin embargo, con el fin de disminuir la impedancia de las líneas y por tanto aumentar la inmunidad a parásitos, usualmente se utiliza una corriente superior por el hecho de establecer una red terminadora de resistencias en el extremo receptor de la línea de transmisión. Ello implica una mayor intensidad solicitada de la línea en el momento de la transmisión del «0», implicando por tanto la utilización de amplificadores de mayor intensidad. Existen múltiples variantes de circuitos integrados comerciales que cumplen esta última premisa.

Trabajando a niveles TTL es posible también compartir una misma línea por distintos emisores y receptores, a condición de utilizar elementos, bien sea cuya salida sea de colector abierto, o bien de 3 estados.

4.7.1.4 *Diferencial*

Aún cuando ya se ha indicado que la modalidad diferencial no es excesivamente utilizada en los métodos de transmisión para entrada/salida en paralelo, no obstante existen aplicaciones en las que la enorme presencia de ruidos en modo común pueden obligar a acudir a esta solución. Existen una serie de distintas normas establecidas entre las cuales se puede contar la EIA RS-422 y la CCITT V-35. Estas normas fundamentalmente están destinadas a la transmisión serie pero pueden ser perfectamente aplicadas a transmisión paralelo en dichos casos especiales. En esencia la modalidad diferencial consiste en la transmisión de una corriente (el carácter distintivo no es la tensión, sino la intensidad), circulante por ambas líneas en serie, en uno u otro sentido, de tal modo que en cada uno de los dos puntos terminales del extremo emisor existen circuitos activos que permiten forzar la aparición de dicha corriente en uno u otro sentido.

4.7.2 Emisores de línea

Se denominan emisores de línea a los circuitos que convierten las señales adecuadas a los niveles lógicos del

circuito electrónico general (normalmente TTL), en los niveles de tensión o corriente adecuados a la línea de transmisión que se haya elegido. Existe una gran diversidad de circuitos ofrecidos por los distintos fabricantes cuya lista sería engorrosa por lo extenso. No obstante se pueden considerar ciertas precauciones de carácter general aplicables a todos ellos, tales como que estos circuitos deben estar situados lo más próximos posibles a la periferia del circuito impreso, de modo que se minimice los caminos de retorno de masa. Las líneas de masa deben ser retornadas hacia la masa propia del circuito integrado y no a la masa del sistema. La salida de una puerta que está emitiendo por una línea no debe ser utilizada como entrada a otros elementos lógicos, a no ser que esta entrada tenga una acción de disparador Schmitt. Deben intercalarse condensadores de desacoplo de la alimentación situados muy próximos a estos circuitos emisores.

4.7.3 Receptores de línea

Las consideraciones anteriores sobre los emisores de línea son totalmente aplicables a los receptores en cuanto a características de compatibilidad con la modalidad elegida para la transmisión, así como en cuanto a la distancia y situación física de dichos circuitos respecto del extremo del circuito impreso y los retornos de masa. Cuando no se utilicen tecnologías especiales (p. ej., cuando se emplean niveles TTL) es muy recomendable la utilización de circuitos con características de histéresis, tal como los disparadores Schmitt con lo que se evitarán problemas debidos a flancos degradados por la línea de transmisión, que pudieran provocar conmutaciones extrañas y por tanto errores en la transmisión de los datos.

4.7.4 Redes terminales

Sea cual sea la modalidad de transmisión elegida es altamente recomendable que las líneas de transmisión dispongan de un circuito terminal cuya impedancia sea equivalente o lo más similar posible a la de la propia línea de transmisión, minimizando de este modo los efectos de reflexiones eléctricas en dicha línea que podían provocar sobreoscilaciones indeseables.

Cuando se utiliza una modalidad de conexión en cadena (Daisy Chain), esta red terminal debe ser situada precisamente en la última posición de la línea de transmisión. En el caso más común de líneas TTL en que utilizan líneas formadas por pares trenzados cuya impedancia está comprendida entre 100 a 200 ohmios, el método más usual consiste en emplear redes de resistencia formadas por una resistencia desde la línea hacia +5 de valor igual a 220 ohmios, y otra resistencia desde la línea a masa, de valor igual a 330 ohmios (fig. 4.43). Aplicando el teorema de Thevenin, esto representa una impedancia terminal de 132 ohmios útil para la mayor parte de aplicaciones. Esta red terminal presenta asimismo la ventaja que en el nivel «1» se obtiene una tensión aproximada de 3 voltios en el extremo receptor, garantizando por tanto dicho nivel, mientras que cuando el extremo emisor transmite un «0» éste domina la situación precedente. Debe tenerse presente también que, si estamos trasmitiendo, tal como se ha indicado anteriormente, a niveles TTL, el circuito emisor debe ser capaz de entregar una salida saturada cuando emite un nivel «0» con una resistencia de 220 ohmios a positivo, lo cual implica que debe ser capaz de saturar con una carga equivalente de 23 miliamperios, normalmente superior al máximo permitido

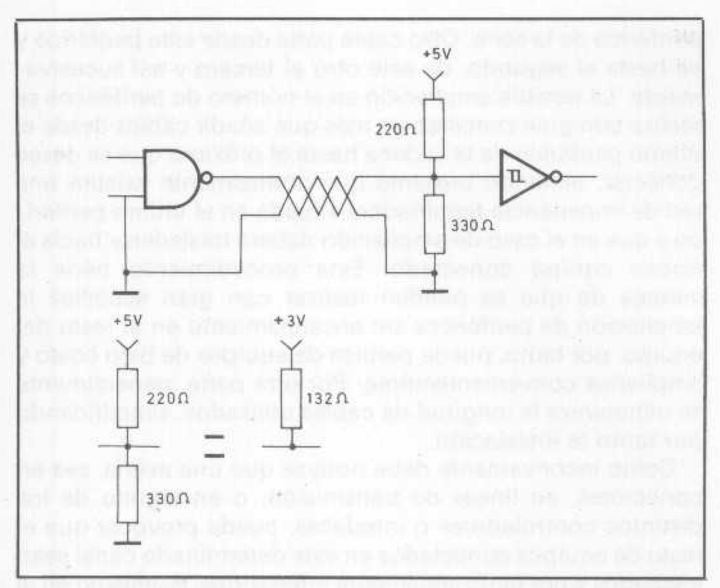


Figura 4.43 Red terminal para conexión de TTL por medio de pares trenzados.

por las puertas normales TTL. Existen múltiples circuitos en la serie 74 que permiten esta corriente con comodidad.

4.7.5 Conectores

Con el fin de permitir una fácil conexión y desconexión de las líneas de transmisión en el extremo de dichas líneas, justo en la entrada del mueble de los equipos, sean unidad central o periféricos, se utilizan conectores multicontacto que permiten una rápida y cómoda instalación. La práctica industrial ha establecido una cierta normalización en cuanto al tipo de conectores y a las señales empleadas, por ejemplo en el caso de la norma V-28, asociada al protocolo V-24, está universalmente aceptado el empleo del conector tipo delta subminiatura de 25 contactos, situando el conector hembra en el extremo periférico, y el macho en la manguera de conexión. En este tipo de conector la norma ISO ha establecido la distribución de señales en cada terminal. Asimismo en el caso de utilización de transmisiones por el bus IEEE-488 se acostumbran a emplear cables terminados en un doble conector macho-hembra, de forma tal que la expansión en cadena es fácilmente realizada a partir del propio conector que llega a cada elemento terminal [9].

4.8 PERIFÉRICOS LOCALES

El concepto de periféricos locales se aplica a los periféricos situados por su propia naturaleza en gran proximidad con la unidad central de proceso. Entre estos periféricos se pueden considerar elementos tales como circuitos integrados específicos para determinadas funciones, como son: ampliación de memoria de acceso directo, memorias de sólo lectura, memorias programables, temporizadores, circuitos de reloj para interrupción, procesadores aritméticos auxiliares, interruptores y pilotos de preselección y visualización de estados, y visualizadores de caracteres de tipos muy variados, pero que generalmente expresan el estado de determinados registros internos del equipo.

4.8.1 Memorias de lectura/escritura (RAM)

En la conexión de memorias RAM debe distinguirse en primer lugar si se trata de memorias estáticas o dinámicas. En

cualquiera de ambos casos existen una serie de señales comunes tales como bits de direccionamiento dependientes de la capacidad individual de cada circuito, cuyos valores de pesos más altos deben ser decodificados para dar lugar a la señal de selección de circuito específico. Por otro lado existen unas señales específicas que deben ser generadas «in situ», tales como la señal R/W («READ/WRITE») que establece la selección de función en base a si se desea efectuar una lectura o bien una escritura. Asimismo existe una señal, cuya denominación es distinta según los fabricantes, que en caso de lectura o bien de escritura permite la selección de los circuitos de salida, permitiendo de este modo la actividad de los pasos de 3 estados en lectura.

En el caso de memorias dinámicas deben asimismo preverse los circuitos apropiados para la regeneración periódica de la información, teniendo presente que debe efectuarse una exploración completa de todas las posiciones cada 1 o 2 milisegundos, dependiendo del circuito empleado. Este circuito de refresco o regeneración debe estar constituido por un contador de direcciones, multiplexado con las direcciones procedentes de la unidad central y que en los ciclos muertos gana acceso a memoria permitiendo los citados ciclos de refresco. Modernamente, y fundamentalmente en memorias de alta capacidad, (de 4 K, 16 K, y superiores) se acostumbran a multiplexar las direcciones de acceso por medio de dos señales de selección RAS y CAS («Row Address Strobe/Column Address Strobe» = validación de dirección de fila/validación de dirección de columna); estas dos señales permiten un fácil multiplexado de la dirección reduciendo el número de bits precisos a la mitad; por tanto son compartidas dichas vías entre la selección de filas y columnas mediante los registros internos a los circuitos. Evidentemente los circuitos periféricos para control de RAM deben ser diseñados específicamente basados en el tipo de circuitos empleados.

4.8.2 Memorias de sólo lectura (ROM)

Respecto a las memorias ROM son de general aplicación las consideraciones anteriormente enunciadas respecto a las memorias RAM. La diferencia fundamental estriba en el hecho de que las memorias ROM solamente son de lectura, y por tanto no es preciso prever ningún tipo de circuitos para la escritura. Asimismo no existe la diversidad de circuitos en cuanto a versiones dinámicas y estáticas, puesto que estas memorias siempre actúan bajo la modalidad estática.

4.8.3 Memorias de sólo lectura programables (PROM y EPROM)

Las memorias PROM y EPROM actúan en cuanto a su lectura de modo similar a las memorias ROM. Se distinguen de las anteriores en que permiten ser programadas por parte del usuario, y en particular las EPROM aceptan la posibilidad de ser borradas para reprogramación. Existen dos tipos fundamentales: memorias en tecnología bipolar consistentes en una matriz de enlaces fusibles, las cuales una vez grabadas son absolutamente inmunes contra agentes externos, y la modalidad formada bajo el concepto de tecnología MOS cuya característica fundamental es que son borrables mediante radiaciones ultravioletas y por tanto deben ser protegidas contra la incidencia accidental de este tipo de radiaciones mediante unos sellos opacos generalmente suministrados por el fabricante.

Tanto las memorias ROM como las PROM y las EPROM deben disponer en cuanto a sus circuitos periféricos de un registro de direcciones seleccionadas, cargado desde la unidad central. En ciertas ocasiones es conveniente añadir un registro de datos de lectura que permita una lectura asíncrona.

4.8.4 Temporizadores

Los circuitos temporizadores programables consisten fundamentalmente en uno o varios contadores cuyo módulo es programable desde la unidad central y que son excitados sea por el reloj general del sistema o bien por relojes específicos. Estos circuitos resuelven uno de los problemas más comunes en cualquier sistema con microprocesadores: la generación de retardos precisos bajo control de programa. En lugar de establecer bucles de programa, el programador simplemente configura este circuito para cubrir sus solicitudes, lo inicializa con la cantidad deseada, y seguidamente el circuito cuenta hasta cubrir el retardo previsto. Posteriormente genera una interrupción hacia la unidad central indicando que ha finalizado su tarea. Fácilmente se puede observar que la sobrecarga de instrucciones en el programa es mínima y que fácilmente se pueden obtener distintos temporizadores simplemente por diversos mandatos. Estos circuitos existen en múltiples variantes dependiendo de los distintos fabricantes y su interfaz acostumbra a ser muy simple por cuanto están diseñados para microprocesadores específicos, incorporando las señales de sincronización, registros de datos y aceptación del bus de direcciones.

4.8.5 Reloj de tiempo real

Se denomina reloj de tiempo real a un circuito, controlado generalmente por cuarzo, que interrumpe periódicamente a la unidad central indicándole la hora presente. Esta interrupción puede ser fija o bien programable, definiendo el intervalo que media entre cada una de ellas. La versión más simple se limita a efectuar una interrupción y un contador de programa lleva el cómputo de la hora real. Otras versiones más elaboradas permiten efectuar interrupciones a distintos niveles según el intervalo que se considere, emitiendo simultáneamente la hora presente, el tiempo de ejecución desde un determinado momento u otras indicaciones horarias que sean precisas. Los comentarios aplicables a los temporizadores en el apartado anterior son también a considerar en este caso en cuanto a los diseños específicos por parte de los fabricantes para simplificar su interfaz.

4.8.6 Procesadores aritméticos auxiliares

Cuando se precisan funciones aritméticas de alto nivel que consumen un gran tiempo y longitud de programa, es conveniente acudir a la utilización de circuitos periféricos calculadores auxiliares.

Un ejemplo es el Am 9511 producido por Advanced Micro Devices [10]. Este circuito, además de las 4 funciones básicas aritméticas (suma, resta, multiplicación y división), puede realizar cálculos trigonométricos e inversamente trigonométricos, raíces cuadradas, logaritmos y exponenciación y puede asimismo almacenar constantes tales como π y **e**. Su modo de operación puede ser tanto con coma fija de simple o doble precisión (16 o 32 bits) o con coma flotante, con precisión única de 32 bits.

Sin circuitos procesadores aritméticos externos, algunas operaciones (incluso de características muy simples tales como la multiplicación) pueden consumir una gran cantidad

de tiempo para la mayor parte de los microprocesadores que realizan esta operación por un sistema de suma y decalaje. Algunos microprocesadores, como por ejemplo el TMS 9900, presentan facilidades para la realización de multiplicación por una única instrucción, pero para realizar cálculos más complejos debe acudirse a rutinas de programa. En líneas generales estos circuitos toman los datos desde el microprocesador, realizan la función programada, y devuelven el resultado a través de un bus de datos bidireccional de 8 bits. Estos circuitos pueden ser conectados al sistema por medio de un puerto o vía de acceso programable de entrada/salida convencional o bien a través de un controlador de acceso directo a memoria que permite una operación mucho más rápida.

4.8.7 Interruptores de preselección

Hoy en día los interruptores de preselección situados en el panel principal de mandos del calculador están perdiendo popularidad debido a la utilización masiva que se hace de los terminales interactivos con el operador, pero no obstante, siguiendo la herencia de los minicomputadores, son utilizados en múltiples ocasiones para poder introducir datos o instrucciones manualmente por simple preselección. Generalmente van asociados a un transceptor de bus, de tal modo que el dato es preseleccionado en la entrada de dicho transceptor, y una vez completado se da una instrucción de transferencia que generalmente lanza una interrupción hacia la unidad central

4.8.8 Pilotos

Las consideraciones anteriormente enunciadas respecto a los interruptores de preselección son también aplicables a los pilotos visualizadores de los estados lógicos de distintos bits, generalmente de los buses de datos y dirección. Cuando deseen visualizarse estos estados, generalmente se colocan registros de múltiples bits (en múltiplos de 4 o de 8) que son activados por medio de una instrucción de salida; en dicho momento el estado del bus deseado (sea el de datos o el de direcciones) es cargado en dichos registros. Como carga de estos registros existen amplificadores (normalmente en forma de circuitos integrados) que atacan directamente a pilotos incandescentes o diodos electroluminiscentes (LED).

4.8.9 Visualizadores

Modernamente los visualizadores de caracteres son mucho más empleados que los pilotos indicadores de bits, por cuanto no es preciso realizar la asociación mental en grupos de bits con el fin de obtener la palabra octal o hexadecimal que comporta su indicación. Existe una gran multiplicidad de tecnologías y modalidades entre los que se cuentan los visualizadores numéricos (en 7 segmentos) y los visualizadores alfanuméricos, bien sean en matriz de 5×7 puntos o bien en 16 o 18 segmentos. Pueden estar construidos a partir de tecnologías muy diversas entre las que se cuentan la de los diodos LED, cristales líquidos LCD, tubos fluorescentes, displays de plasma o tubos indicadores de alta tensión.

En cualquiera de estos casos consisten básicamente en un registro de 4 u 8 bits asociado a cada carácter cargado desde la unidad central por una instrucción de salida. Los diversos bits de salida de este registro son aplicados a través de un multiplexor a una memoria ROM generadora de caracteres, y a su salida sendos amplificadores específicos al tipo de tecnología utilizada realizan el ataque a los diversos puntos o segmentos. La misión del multiplexor del intercalado es hacer innecesario realizar múltiples copias del circuito generador de caracteres y reducir asimismo el número de amplificadores de ataque a los distintos elementos activos. Este multiplexor está generalmente controlado por un reloj de baja frecuencia (pero superior a 50 Hz para evitar batidos visuales) que va realizando la exploración de todo el visualizador a una cierta cadencia para presentar la información aparentemente en paralelo.

4.9 PERIFÉRICOS REMOTOS

Se consideran periféricos remotos a aquellos que están situados generalmente fuera del conjunto contenedor de la unidad central. Están ligados a ella por medio de una línea de transmisión con sendos conectores en sus extremos y disponen de circuitos electrónicos específicos por cada terminal. Por el extremo correspondiente a la unidad central se establecen los circuitos de interfaz apropiados al microprocesador que se utilice, mientras que por el extremo asociado al periférico se incluye usualmente un controlador (muy a menudo incluyendo algún microprocesador específicamente dedicado a esta función) que permite traducir las instrucciones de la unidad central en las funciones precisas para la actuación electrónica o electromecánica del equipo de que se trate. Dada la gran diversidad de equipos existentes y las múltiples variantes dentro de cada uno (que serán tratadas en profundidad en sucesivos capítulos de esta misma obra) solamente se da una relación de los tipos de periféricos más usuales que pueden encontrarse asociados a sistemas de microprocesadores.

Los periféricos más usuales son:

—Teclados
—Teletipos
—Impresores seriales
—Impresores de líneas
—Visualizadores en TRC
—Cassettes
—Cinta magnética
—Disco flexible
—Disco magnético
—Lectores de tarjeta
—Perforadores de cinta
—Lectores de cinta
—Microprocesadores «esclavos»
—Instrumentos de medida
—Sensores digitales
—Sensores analógicos
—Actuadores digitales
—Generadores de señales analógicos
—Fuentes de alimentación programables.

Dada la proporción de electrónica que representan los circuitos de interfaz y controladores de periféricos, respecto al total del sistema, estos circuitos serán los que requieran mayor atención del técnico tanto si es a nivel del diseño, como de adaptación de circuitos comerciales, como de mantenimiento.

En cualquier caso deben observarse atentamente las instrucciones relativas a: relaciones de señales, tiempos y niveles, protocolos, y todas las observaciones que el fabricante del equipo electromecánico haya especificado.

REFERENCIAS

[1] R. M. Grossman. EDN nov. 78
[2] MCS-85 Users Manual. Intel. 1978
[3] Bipolar microcomputers data book. Texas instr. 1977
[4] TTL data book. Texas instr. 1976
[5] F8 microprocessor devices. Mostek. 1976
[6] Micronova data manual. Data general. 1977
[7] Oem Users Manual. Pertec. 1972
[8] Line drivers and receivers. Texas instr.
[9] C. A. Ogdin. Mini-micro systems. Jul-78
[10] A. Santini. Microprocessor data manual. Hayden. 1978

entradas-salidas en serie

5

Victor Gil Galés

5.1 INTRODUCCION

La transferencia de información entre dos sistemas digitales, por ejemplo, un microcomputador y un terminal, periférico u otro microcomputador, se realiza generalmente carácter a carácter utilizando códigos binarios (ASCII, EBCDIC, BAUDOT,...). Otras veces la información que se transfiere no corresponde a ninguna codificación de caracteres numéricos ó alfanuméricos sino que es puramente binaria, por ejemplo, cuando se efectúan cargas de programas objeto sobre la memoria del ordenador.

De una forma o de otra la información se transmite en unidades de información denominadas palabras, que suelen ser de 5 a 8 bits. Existen dos formas de realizar la transmisión de estas palabras:

— *Método paralelo:* Transmitiendo simultáneamente, por líneas separadas, todos los bits de la palabra, junto con una señal de reloj que indica el momento en que está presente una palabra de información en las líneas de datos (fig. 5.1).

— *Método serie:* Transmitiendo en forma secuencial en el tiempo todos los bits de la palabra, uno tras otro, por una sola línea de datos.

Eventualmente puede existir una línea adicional de reloj que marca los tiempos de bit (fig. 5.2).

El método paralelo es utilizado para transmisiones a alta velocidad entre dos sistemas; no obstante cuando la distancia entre ambos aumenta, el coste de la línea y el de los amplificadores de transmisión y recepción puede llegar a crecer de forma tal que, desde el punto de vista económico, sea preferible utilizar un sistema serie de comunicaciones.

Por otra parte, como se verá más adelante, los sistemas de comunicaciones serie han alcanzado un alto grado de estandarización. Existen normas universalmente aceptadas que fijan completamente todos los detalles de la comunicación, incluyendo aspectos mecánicos (tipo de conector y distribución de señales en las patillas del mismo), aspectos eléctricos (niveles y formas de las señales) y aspectos lógicos (sistemas de codificación y sincronización, y descripción de todos los circuitos de datos, control y temporizado).

Estos estándares han conducido a que la mayoría de fabricantes de procesadores y periféricos incorporen en sus equipos interfaces serie que cumplen las normas especificadas, de forma que se pueda realizar con toda facilidad la conexión indistinta de cualquier terminal o periférico con cualquier procesador.

Así, se utilizan interfaces serie para conectar periféricos, como terminales de pantalla o impresoras, a computadores aunque su distancia sea reducida y puedan, por tanto, usarse interface de tipo paralelo.

Por último, un tercer campo en que se utilizan sistemas de manipulación de datos en serie es el de los controladores de unidades de almacenamiento de informaciones digitales sobre soportes magnéticos (discos, cassettes y diskettes). En ellos se graban y se leen los datos en forma serie, presentándose problemas de codificación comunes con los sistemas de comunicaciones serie.

El objeto del presente capítulo es hacer un estudio detallado de los sistemas de transmisión de datos en serie. Se describirán los métodos de codificación, los estándares que regulan las comunicaciones en serie, los componentes que existen en el mercado para la realización de sistemas de comunicaciones y finalmente se presentarán unos ejemplos prácticos de utilización de los mismos.

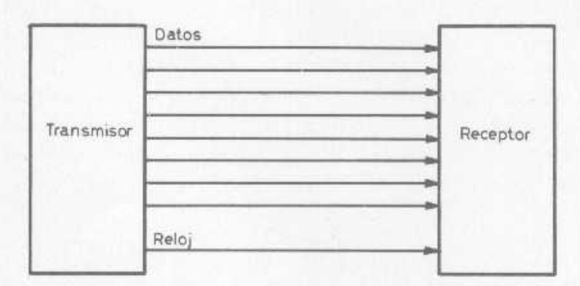


Figura 5.1 Transmisión paralelo

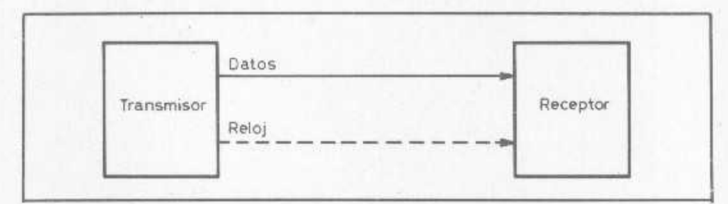


Figura 5.2 Transmisión serie

5.2 PROBLEMAS EN LAS TRANSMISIONES SERIE

Cuando se transmiten informaciones a través de una línea serie es necesario utilizar un sistema de codificación que permita resolver los siguientes problemas:

— *Sincronización de bit.* El receptor necesita saber exactamente donde empieza y donde termina cada bit en la señal recibida para efectuar el muestreo de la misma en el centro de la celda de bit.

Considérese el caso de transmisión en serie de la información 01110010. Si se utilizase un método NRZ (no retorno a cero, en que los bits 1 ó 0 se representan por niveles 1 ó 0 respectivamente), la señal en la línea sería como la representada en la figura 5.3.

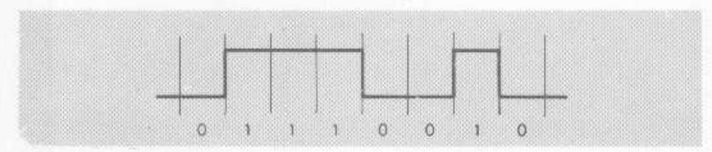


Figura 5.3 Codificación NZR

La presencia de varios bits iguales, por ejemplo 3 «unos», hace que la línea no efectúe ninguna transición y el receptor puede perder la pista de donde empieza y donde acaba cada bit.

Si el número de bits iguales aumenta, se observa que la dificultad de reconstruir las celdas de bit aumenta también.

— *Sincronización de carácter.* La información en serie se transmite, por definición, bit a bit, pero la misma tiene sentido en palabras, por ejemplo de 8 bits.

El sistema de codificación usado debe permitir distinguir sin ambigüedades dentro de una corriente de bits cuáles son los 8 que forman una palabra.

Por ejemplo, la siguiente información en serie:

010011000100110 0100

Puede tener distintas interpretaciones según como se agrupen los 8 bits para formar las palabras.

01	00110001	00110010
010	01100010	01100100

La primera agrupación representa los caracteres 1 y 2 según una codificación ASCII, la segunda representa, según la misma codificación, los caracteres *b* y *d*.

— *Sincronización de mensaje.* En un sistema de comunicaciones generalmente las informaciones se transmiten en bloques de caracteres.

Por sincronización de mensaje entendemos el mecanismo por el cual un conjunto de palabras es interpretado correctamente. Este problema normalmente no incumbe a los circuitos de codificación, sino al procesador que lo utiliza.

El conjunto de reglas (protocolo) que permiten interpretar correctamente los mensajes suele estar controlado por una tarea software (un programa) que ejecuta el ordenador, aunque actualmente hay ciertos circuitos integrados LSI que efectúan alguna de estas tareas.

En los siguientes apartados se trata con más detalle de los sistemas de sincronización de bit y carácter, en sistemas de señales serie en general. En el apartado 5.3 se estudian concretamente los sistemas de comunicaciones - serie.

5.2.1 Sincronización de bit

Para resolver el problema de la sincronización de bit pueden usarse varios métodos:

a) Enviar por una línea independiente de la de datos una señal de reloj que indique el centro de las celdas de bits de la línea de datos.
b) Enviar junto con cada bit transmitido en serie y por la misma línea una información adicional que permita al receptor extraer una señal de reloj.
c) Lograr mediante algún procedimiento que los relojes de transmisión y recepción se mantengan en fase continuamente.

Para ilustrar con mayor claridad estos conceptos veamos un proceso de transmisión y recepción con algún detalle.

— Se desea transmitir 8 bits en serie, por ejemplo 01110010.

— El transmisor, en cada flanco de subida de un reloj, envía un bit de información por la línea.

— Las señales por la línea, a lo largo del tiempo, contienen unas celdas de bit, que dependen de la frecuencia del reloj, y dentro de cada celda, el transmisor coloca un bit

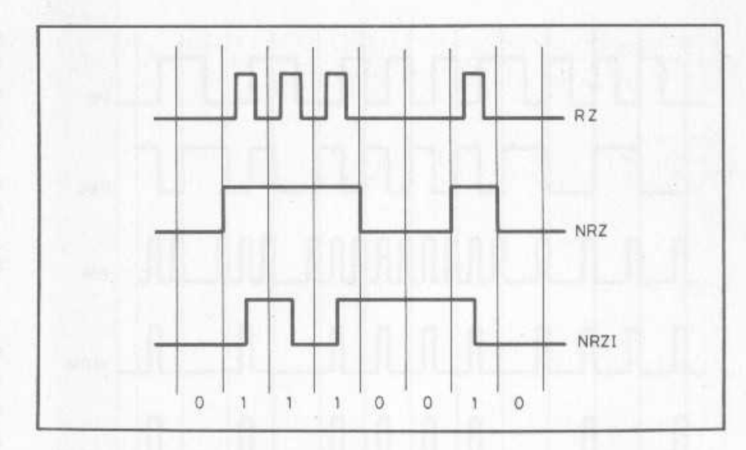


Figura 5.4 Codificaciones no auto-reloj

de información codificado según algún procedimiento, por ejemplo:

RZ. Una celda de bit contiene un 1 si hay un impulso positivo y un 0 si no lo hay.

NRZ. La celda de bit contiene un 1 o un 0 según el nivel de la señal sea 1 ó 0.

NRZI. La celda de bit contiene un 1 si hay una transición y un 0 si no la hay (fig. 5.4).

Para que el receptor pueda interpretar adecuadamente estas señales, debe ser capaz de obtener o crear un reloj que se mantenga en perfecto sincronismo con el del transmisor. Este reloj marcará las celdas de bit y analizándolas verá si contienen un bit 1 ó 0.

Como puede observarse en la figura 5.5, los datos no contienen información de reloj.

Efectivamente, las secuencias de ceros, en cualquiera de los sistemas (RZ, NRZ, NRZJ), y las secuencias de unos, en el sistema NRZ, no contienen ninguna transición que permita al receptor determinar la situación de las celdas de bit.

Estos sistemas se dice que son *no self-clock* (es decir sin auto-reloj). La sincronización de bit en tales sistemas se consigue utilizando en la recepción el propio reloj de transmisión, enviado por una línea independiente de los datos, o bien utilizando relojes de precisión y con dispositivos adicionales que aseguren que se mantiene a la misma frecuencia y fase que el de transmisión. Alguno de estos sistemas de reloj se describe más adelante (apartado 5.3.3).

Frente a estos sistemas de codificación se encuentran los de auto-reloj *(self-clock)*, que transmiten información de forma tal que permiten al receptor deducir la situación exacta de las celdas de bit y por tanto los datos, sin necesidad de disponer de un reloj síncrono con el de transmisión.

Hay varios métodos auto-reloj, siendo los más conocidos

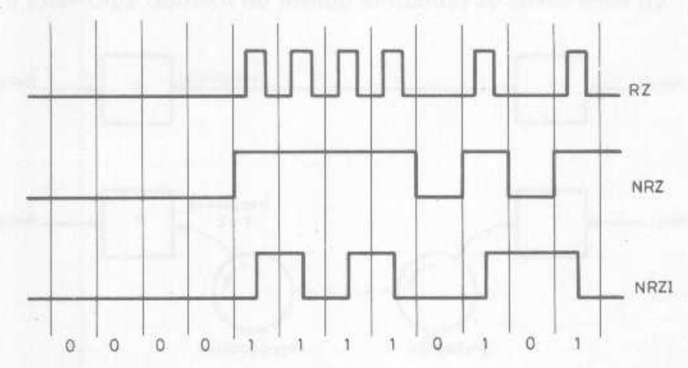


Figura 5.5 Codificaciones no auto-reloj

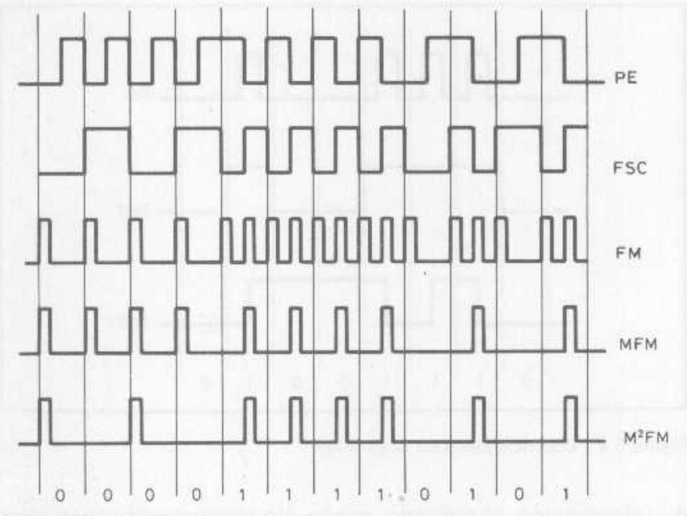


Figura 5.6 Codificaciones auto-reloj

los utilizados en la figura 5.6: PE, codificación de fase; FSC, codificación por cambio de frecuencia; FM, modulación de frecuencia; MFM y M^2FM, modulación de frecuencia modificadas.

En estos sistemas, el envío de la información adicional para determinación del reloj se hace a costa de la disminución de la cantidad de información útil enviada para un mismo ancho de banda.

En efecto, tal como se observa en las figuras 5.5 y 5.6, los métodos «no auto-reloj» permiten enviar una misma información a una frecuencia mitad que en los métodos «auto-reloj».

Dado que las características de una línea o canal de transmisión limitan la frecuencia máxima de la señal que se puede enviar por él, la cantidad que es posible enviar mediante una codificación «no auto-reloj» es doble que mediante una codificación «auto-reloj». No obstante hay campos de aplicación idóneos para cada método.

Cuando el problema es de transmisión de una información serie entre dos puntos, es posible la utilización de una codificación «no auto-reloj», realizando la sincronización de bit con el propio reloj de transmisión o generando un reloj sincronizado con aquél.

Cuando el problema es de grabación de información serie en un soporte magnético giratorio (discos, cintas, etc.), para posterior reproducción o lectura, la posibilidad de utilizar el reloj usado en la grabación, o sincronizar un reloj de recepción, se hace muy difícil al introducirse un agente perturbador como es el de las fluctuaciones de las velocidades de giro del soporte magnético en los instantes de grabación y lectura.

En tales casos es preferible utilizar un método auto-reloj y

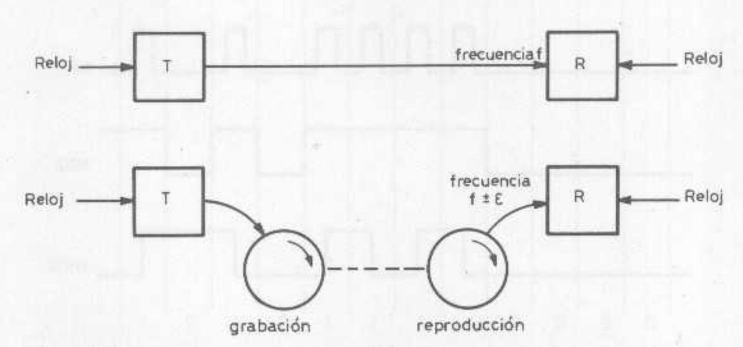


Figura 5.7 Sistemas serie de transmisión y de grabación/reproducción

disponer en el receptor un circuito que extraiga el reloj de recepción de los datos, haciéndose insensible a posibles derivas del reloj de grabación y de las velocidades del soporte (fig. 5.7).

Los métodos de sincronización de bit en los sistemas «no auto-reloj» para comunicaciones serie se analizan con mayor detalle en el apartado 5.3.

5.2.2 Sincronización de carácter

Para obtener la sincronización de carácter pueden utilizarse diversos sistemas, unos se basan en la utilización de líneas adicionales a las de datos para enviar impulsos que indican el inicio de un bloque de caracteres. Tal impulso identifica el primer bit del primer carácter de un bloque o mensaje, y luego, por contaje de bits y caracteres se determinan todas las fronteras de los datos del bloque.

Otros sistemas, utilizados usualmente en los sistemas de comunicaciones serie, son:

— *Asíncrono:* Cada carácter va señalizado mediante dos bits, uno al principio, bit de arranque, y otro al final, bit de parada.

Estos bits permiten reconocer las fronteras de los caracteres.

— *Síncrono:* Cada mensaje o bloque de transmisiión va precedido por unos caracteres de sincronismo.

Cuando el receptor identifica una configuración de bits igual a la de los caracteres de sincronismo, da por detectado el inicio de los datos y a continuación, por contaje de bits y caracteres identifica todos los caracteres del bloque.

5.3 METODOS DE ENTRADA/SALIDA SERIE PARA COMUNICACIONES CON TERMINALES

En lo que sigue se utiliza el nombre genérico de terminales para designar a los sistemas que se comunican utilizando un procedimiento serie de entrada/salida. Un terminal puede ser un ordenador, un microcomputador, un periférico, etc.

La comunicación entre terminales se hace utilizando líneas o canales de transmisión, que pueden ser:

— *Simplex,* cuando son capaces de transmitir información en un solo sentido.

— *Semiduplex* (half-duplex), cuando son capaces de transmitir información en ambos sentidos pero no de forma simultánea.

— *Dúplex* (full-duplex), cuando son capaces de transmitir simultáneamente información en ambos sentidos.

La codificación de las señales en estos sistemas se hace mediante uno de los siguientes métodos: asíncrono o síncrono.

5.3.1 Método asíncrono

En el método asíncrono la transmisión se controla por bits de inicio y de final que enmarcan cada carácter transmitido, son los denominados bits de start y stop y son utilizados por el terminal receptor para sincronizar su reloj con el del transmisor en cada carácter.

La especificación RS404 de EIA (Electronic Industries Association) define las características del método asíncrono de transmisión serie.

La transmisión en asíncrono se basa en las siguientes reglas:

a) Cuando no se envían datos por la línea, ésta se mantiene en estado 1.

b) Cuando se desea transmitir un carácter se envía primero un bit de inicio, que pone la línea a cero durante el tiempo de 1 bit.
c) A continuación se envían todos los bits del carácter a transmitir con los intervalos que marca el reloj de transmisión.
d) A continuación del último bit del carácter se envía el bit de final que hace que la línea se ponga a 1 por lo menos durante el tiempo de 1 bit.

Los datos codificados según estas reglas pueden ser detectados fácilmente por el receptor. Para ello deben seguirse los siguientes pasos:

1) Esperar una transición de 1 a 0 en la señal recibida.
2) Activar un reloj de frecuencia igual a la del transmisor.
3) Muestrear la señal recibida al ritmo de este reloj para formar el carácter.
4) Leer un bit más de la línea y comprobar si es 1 para confirmar que no ha habido error de sincronización.

En las figuras 5.8 y 5.9 se presentan unos organigramas que definen con mayor precisión los métodos de transmisión y recepción en asíncrono.

El bit de final tiene la misión de llevar la línea a estado 1 para que el bit de inicio del próximo carácter provoque la transición de 1 a 0 que permita al receptor sincronizar el siguiente carácter.

El bit de final sirve también para dar tiempo a que el sistema receptor acepte el dato recibido. De todas formas, actualmente se utilizan siempre registros tampón que almacenan el dato recibido mientras el receptor está recibiendo el siguiente, de forma que el procesador dispone del tiempo de todo un carácter para recogerlo.

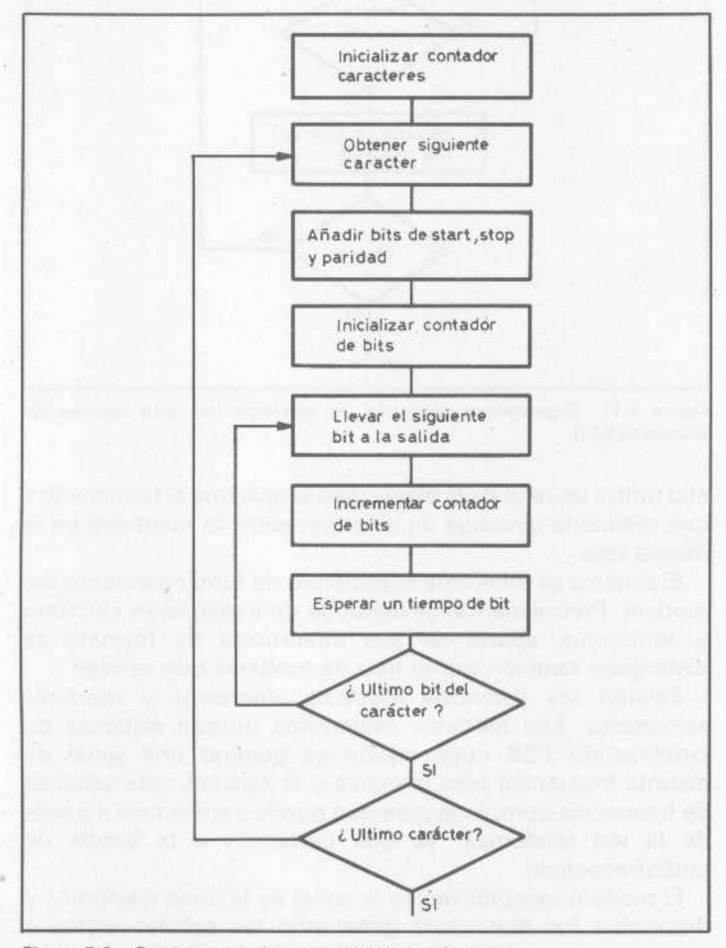


Figura 5.8 Organigrama de transmisión en asíncrono

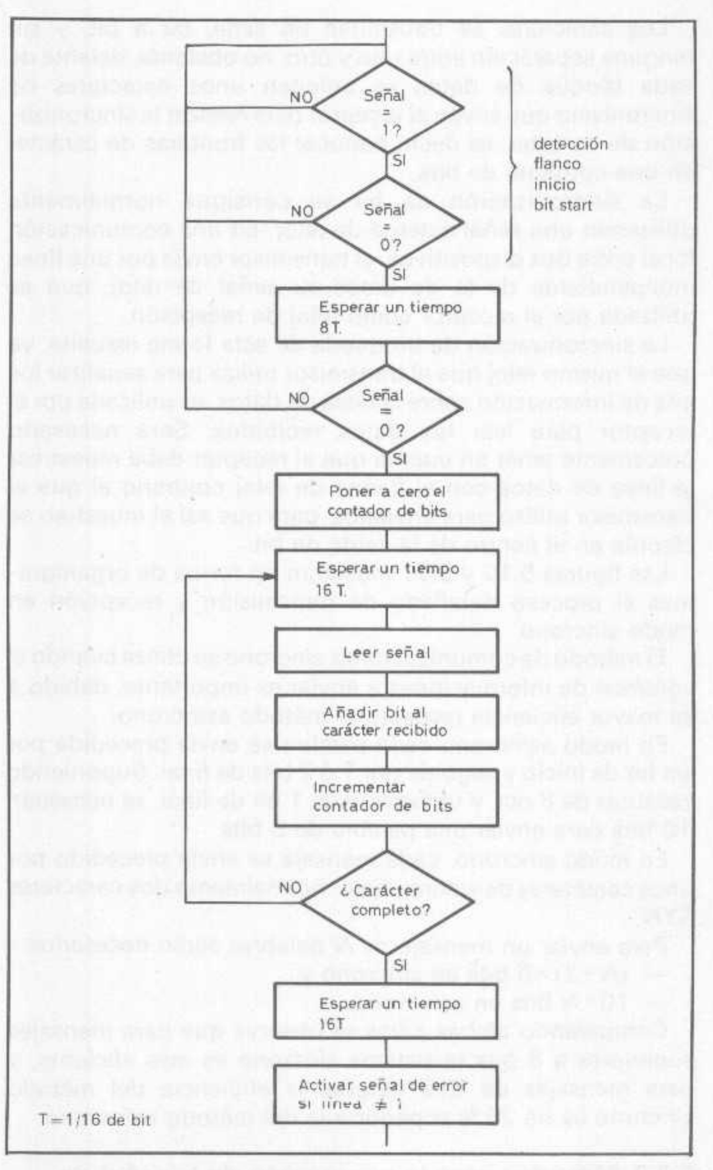


Figura 5.9 Organigrama recepción en asíncrono

El método asíncrono de transmisión presenta las siguientes ventajas:

1) Permite enviar caracteres a ritmos variables ya que cada uno de ellos lleva incorporada la información de sincronismo.
2) Existen circuitos integrados de bajo costo, las UART, que simplifican enormemente la realización de sistemas de entrada/salida en este formato.
3) Es un método de comunicaciones estándar entre ordenadores y terminales de pantalla, impresoras lentas, etc.

Entre sus inconvenientes se puede citar, como más importante, su ineficiencia, ya que cada carácter va lastrado con dos bits de sincronización que no contienen información útil. Asumiendo caracteres de 8 bits, es necesario enviar por la línea 10 bits para enviar un carácter, es decir sólo un 80 % de la información transmitida es válida.

5.3.2 Método síncrono

En el método síncrono, en vez de añadirse bits de sincronismo a cada palabra, lo que se hace es añadir caracteres de sincronismo a cada bloque de datos.

Los caracteres se transmiten en serie, bit a bit, y sin ninguna separación entre uno y otro, no obstante, delante de cada bloque de datos se colocan unos caracteres de sincronismo que sirven al receptor para realizar la sincronización de carácter, es decir, conocer las fronteras de carácter en una corriente de bits.

La sincronización de bit se consigue normalmente utilizando una señal externa de reloj. En una comunicación local entre dos dispositivos, el transmisor envía por una línea independiente de la de datos su señal de reloj, que es utilizada por el receptor como reloj de recepción.

La sincronización de bit queda de esta forma resuelta, ya que el mismo reloj que el transmisor utiliza para serializar los bits de información sobre la línea de datos, es utilizada por el receptor para leer los datos recibidos. Será necesario únicamente tener en cuenta que el receptor debe muestrear la línea de datos con el flanco de reloj contrario al que el transmisor utilizó para enviarlos, para que así el muestreo se efectúe en el centro de la celda de bit.

Las figuras 5.10 y 5.11 muestran en forma de organigramas el proceso detallado de transmisión y recepción en modo síncrono.

El método de comunicaciones síncrono se utiliza cuando el volumen de informaciones a enviar es importante, debido a su mayor eficiencia respecto al método asíncrono.

En modo asíncrono cada palabra se envía precedida por un bit de inicio y seguida por 1 ó 2 bits de final. Suponiendo palabras de 8 oct. y utilización de 1 bit de final, se necesitan 10 bits para enviar una palabra de 8 bits.

En modo síncrono, cada mensaje se envía precedido por unos caracteres de sincronismo, normalmente dos caracteres SYN.

Para enviar un mensaje de N palabras serán necesarios:

— $(N+2)\times 8$ bits en síncrono y

— $10\times N$ bits en asíncrono.

Comparando ambas cifras se observa que para mensajes superiores a 8 bits el sistema síncrono es más eficiente, y para mensajes de 512 octetos la eficiencia del método síncrono es un 25 % superior a la del método asíncrono.

5.3.3 Métodos para la generación de la señal de reloj

En una comunicación remota utilizando modems la señal de reloj es extraída del canal de datos por el modem; para ello utiliza un reloj de la misma frecuencia que el transmisor y que mediante circuitos de sincronización lo mantiene en la misma fase.

El sistema es inherente al principio de funcionamiento del modem. Precisamente los métodos de transmisión síncrono y asíncrono, aparte de sus diferencias de formato se distinguen también por el tipo de modems que utilizan.

Existen los llamados *modems síncronos* y *modems asíncronos*. Los modems asíncronos utilizan sistemas de codificación FSK cuya misión es generar una señal de distinta frecuencia para la *marca* y el *espacio*, esta señal es de frecuencia apropiada para que pueda transmitirse a través de la red telefónica, ya que pertenece a la banda de audiofrecuencia.

El modem receptor recibe la señal de la línea telefónica y discrimina los dos tonos generando las señales marca y espacio que reconstruyen la señal digital primitiva.

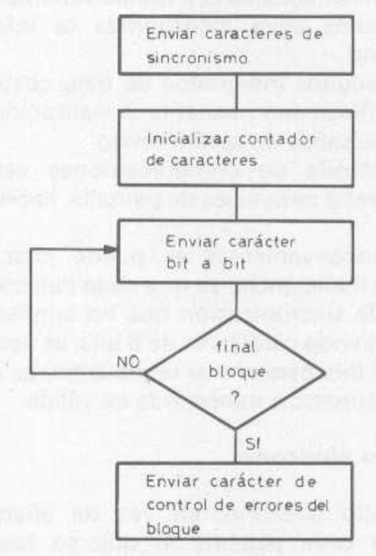


Figura 5.10 Organigrama transmisión en síncrono

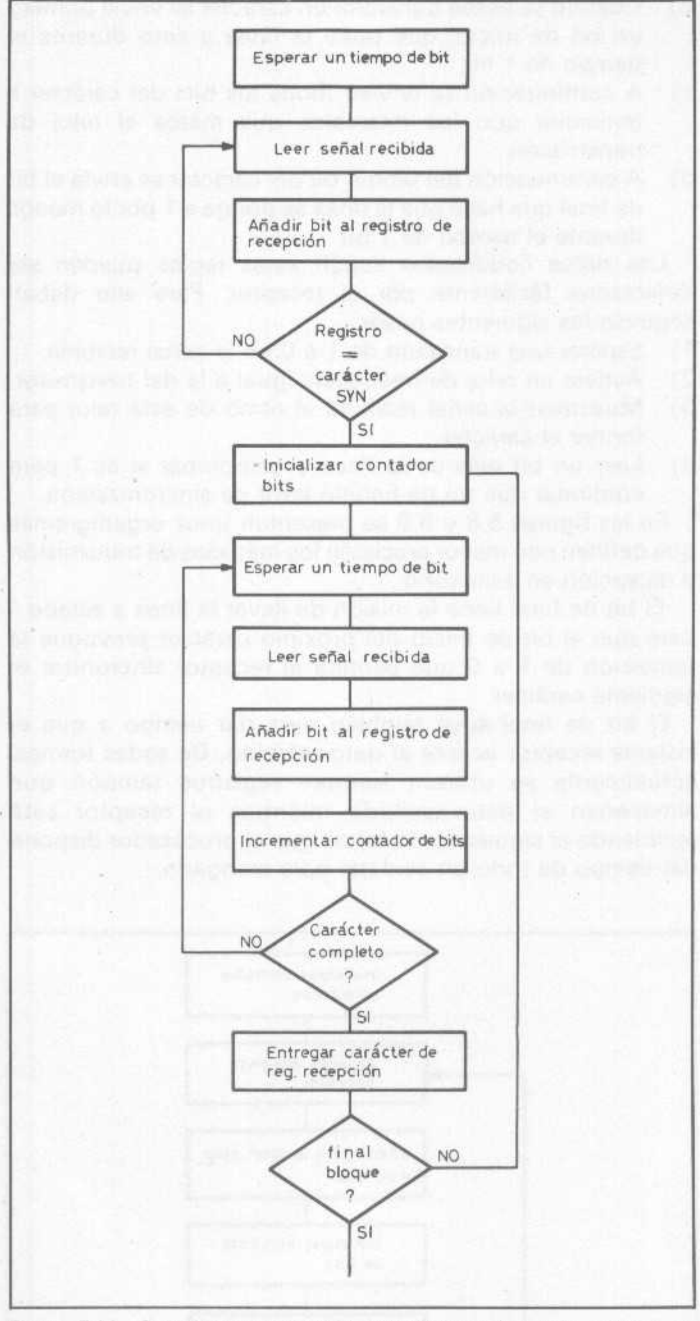


Figura 5.11 Organigrama recepción en síncrono (un solo carácter de sincronización)

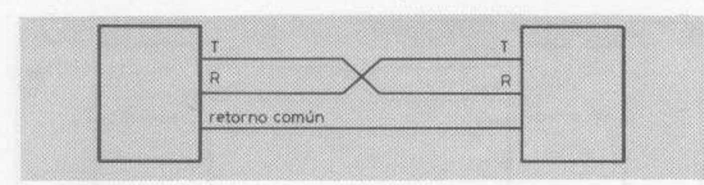


Figura 5.12

Debido a este modo de funcionamiento, el modem en sí no está ligado a la frecuencia de transmisión de los datos y admite, sin necesidad de ningún ajuste, señales de frecuencias de transmisión comprendidas entre cero y el máximo.

En cambio, los modems síncronos utilizan sistemas de codificación por cambio de fase y permiten transmitir mayor cantidad de bit por segundo, es decir, mayores frecuencias de transmisión sobre los mismos canales telefónicos. Para la demodulación de la señal, el receptor dispone de un reloj de la misma frecuencia que el transmisor y lo mantiene en fase con aquél mediante un circuito PLL; la detección de los cambios de fase entre el reloj y los datos le permite reconstruir los datos binarios, obteniéndose además una señal de reloj que marca las fronteras de los bits y que por tanto permite la comunicación utilizando el método síncrono.

Para comunicaciones asíncronas se utilizan normalmente modems asíncronos, pero pueden utilizarse también los modems síncronos, esto permite aumentar la velocidad de transmisión por la línea, al utilizar modems de mayor velocidad, pero sin necesidad de cambiar el protocolo de comunicación. En tal tipo de comunicación, llamada a veces *isosincrona* o *isocrona*, se utiliza el reloj del modem como reloj de recepción x1 o bien puede utilizarse un reloj interno de frecuencia x16 y realizar la sincronización de bit por el método asíncrono.

Para transmisiones síncronas es necesario en principio utilizar modems síncronos, ya que el sistema de recepción necesita el reloj de bit que el modem síncrono genera.

No obstante, se pueden utilizar modems asíncronos para transmisiones síncronas, de la misma forma que pueden realizarse comunicaciones síncronas entre dos terminales sin modems y sin que el terminal receptor reciba el reloj transmisor, es decir una comunicación síncrona con solo 3 hilos, como la indicada en la figura 5.12.

Para ello se utiliza en el receptor un reloj de alta estabilidad y que además se resincroniza con los flancos de las señales de datos.

La relación entre señales de datos y de reloj en el transmisor es la indicada en la figura 5.13.

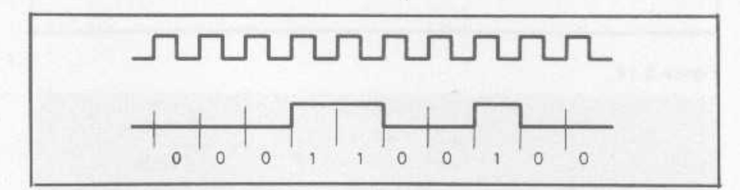


Figura 5.13

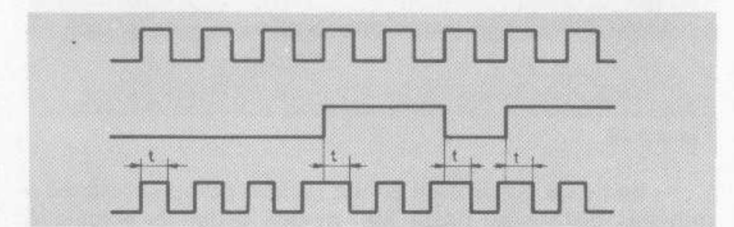


Figura 5.14

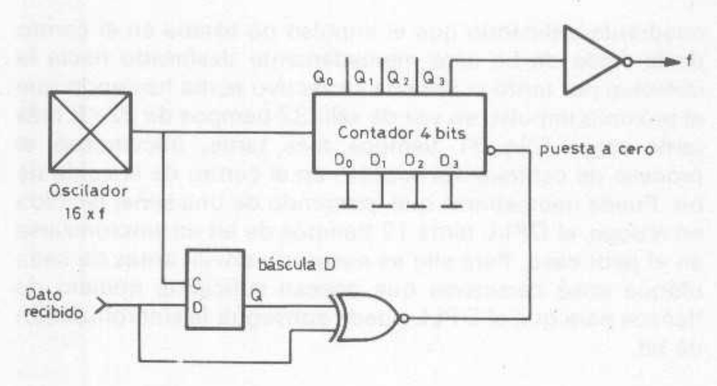


Figura 5.15 Circuito de sincronización del reloj con los datos recibidos

El objetivo es conseguir que el reloj del receptor esté en fase lo más exactamente posible con el reloj del transmisor y para ello se aprovecha cada flanco de la señal recibida para reiniciar el ciclo del reloj de recepción. En la figura 5.14 se ve este efecto de resincronización en el reloj de recepción cuya frecuencia ha derivado y se ha hecho mayor (en forma exagerada) que la del transmisor.

Un ejemplo de tal sistema de reloj es el mostrado en la figura 5.15.

El sistema utiliza un reloj de frecuencia 16 veces la del reloj de transmisión ($16 \times f$).

La salida de peso más alto del contador de 4 bits es el reloj de frecuencia *f*. Esta señal tiene el flanco de subida en el momento en que el contador se pone a 0. La sincronización se efectúa mediante la puesta a 0 asíncrona del contador cada vez que se detecta un flanco en la línea de datos. En el momento del flanco, el contador debe estar a 0; si no lo está, significa que el reloj tiende a adelantarse o atrasarse, y se aprovecha este momento para ponerlo en sincronismo.

El sistema de detección de flancos utiliza una bascula D para generar una salida retrasada 1/16 de bit respecto a la señal de entrada, la puerta NO-O-exclusiva genera un impulso de 0 en los instantes en que la señal *D* varía.

Otro sistema es el basado en un PLL (phase locked loop) digital. Por ejemplo, el 8273, controlador de comunicaciones para protocolos tipo HDLC y SDLC, incorpora un PLL digital que funciona de la siguiente forma:

Se utiliza un reloj de frecuencia 32 veces superior a la de transmisión. A partir de este reloj $32 \times R$ y del reloj recibido, el DPLL genera un impulso que está centrado en las celdas de bit.

El DPLL reacciona contra derivas y distorsiones de fase en los datos recibidos haciendo correcciones en la fase del impulso de reloj a base de incrementos discretos.

El impulso del DPLL se genera a partir de un contaje de 32 impulsos del reloj $32 \times R$, con una corrección de -2, -1, $+1$ ó $+2$ según el cuadrante en que se detecta el flanco de la señal recibida (fig. 5.16).

En este ejemplo el flanco de la señal se detecta en el 2.º

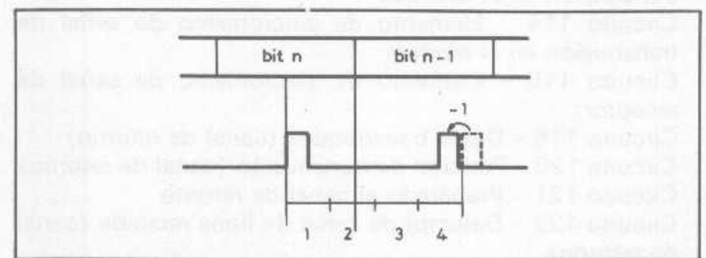


Figura 5.16

cuadrante indicando que el impulso no estaba en el centro de la celda de bit sino marcadamente desfasado hacia la derecha; por tanto el sistema correctivo actúa haciendo que el próximo impulso en vez de salir 32 tiempos de 32×R más tarde salga sólo 31 tiempos más tarde, iniciándose el proceso de centraje del impulso en el centro de la celda de bit. Puede demostrarse que, partiendo de una señal recibida en reposo, el DPLL tarda 12 tiempos de bit en sincronizarse en el peor caso. Para ello es necesario enviar antes de cada bloque unos caracteres que posean suficiente número de flancos para que el DPLL pueda conseguir la sincronización de bit.

5.4 LOS ESTANDARES DE LAS COMUNICACIONES SERIE

Los sistemas de comunicaciones serie tienen a su disposición un conjunto de recomendaciones elaboradas por asociaciones e institutos de normalización (ISO, EIA, CCITT,....) que especifican con precisión todas las características del sistema de comunicaciones.

Las normas para comunicaciones serie están clasificadas por niveles; aquí interesa resaltar únicamente el NIVEL 1, que hace referencia a:

a) Las características eléctricas de las señales
b) Las características mecánicas de la interfaz
c) La descripción funcional de las señales.

La norma más ampliamente aceptada es la EIA RS-232-C, que define las características funcionales, eléctricas y mecánicas de la interfaz entre un terminal y un equipo de comunicaciones (p. ej. un modem). La norma RS-232-C puede ser aplicada a la conexión entre dos terminales, aunque no se utilicen modems, como se verá a continuación.

Las especificaciones funcionales de la RS-232-C coinciden con la recomendación V.24 del CCITT (Comité Consultatif International Telephonique et Telegraphique) y definen 21 circuitos con el siguiente significado. (Se utiliza la numeración de circuitos según CCITT).

Circuito 101 - Tierra de protección
Circuito 102 - Retorno común
Circuito 103 - Dato transmitido
Circuito 104 - Dato recibido
Circuito 105 - Petición de emisión
Circuito 106 - Preparado para emisión
Circuito 107 - Preparado el equipo de comunicaciones
Circuito 108 - Terminal de datos preparado
Circuito 109 - Detector de señal de línea recibida
Circuito 110 - Detector de calidad de señal
Circuito 111 - Selector de velocidad de transmisión en el terminal
Circuito 112 - Selector de velocidad de transmisión en el modem
Circuito 113 - Elemento de sincronismo de señal de transmisión en el terminal
Circuito 114 - Elemento de sincronismo de señal de transmisión en el modem
Circuito 115 - Elemento de sincronismo de señal de receptor
Circuito 118 - Datos transmitidos (canal de retorno)
Circuito 120 - Petición de transmisión (canal de retorno)
Circuito 121 - Preparado el canal de retorno
Circuito 122 - Detector de señal de línea recibida (canal de retorno)
Circuito 125 - Detector de llamada

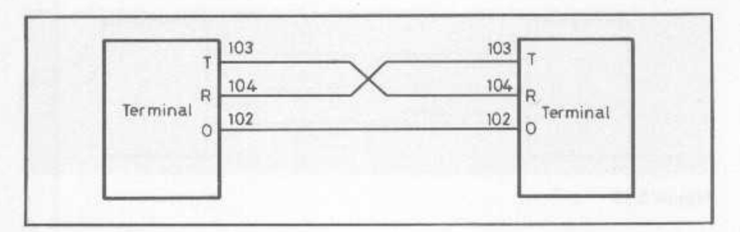


Figura 5.17

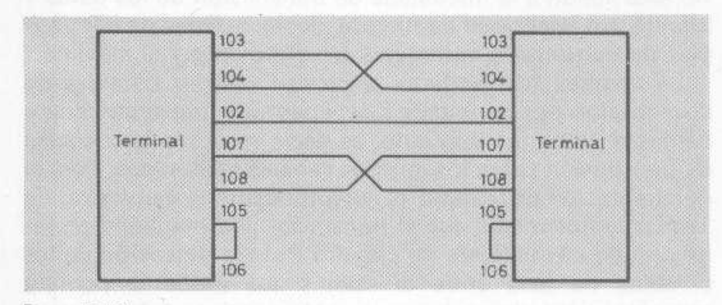


Figura 5.18

Estas son las 21 señales que RS-232-C y V.24 especifican para la comunicación entre un terminal y un modem.

Para la comunicación entre dos terminales, sin utilización de modems, se utiliza un subconjunto de 3, 5 ó 7 señales solamente, aunque se respetan sus especificaciones funcionales, eléctricas y mecánicas. Las figuras 5.17 y 5.18 muestran dos conexiones RS-232-C entre terminales.

En cuanto a especificaciones mecánicas, la norma RS-232-C establece un conector de 25 patillas y fija todas sus dimensiones, así como la distribución de los 21 circuitos en el mismo.

Especificaciones eléctricas:

Para la comunicación de dos terminales a través de una línea serie se utilizan diversas configuraciones de emisores y receptores, las que más se utilizan son las siguientes:

— Señales TTL. Envío directo por una línea unifilar o por pares trenzados de las señales de salida de las puertas TTL.

No es aconsejable su utilización para distancias mayores de 5 metros.

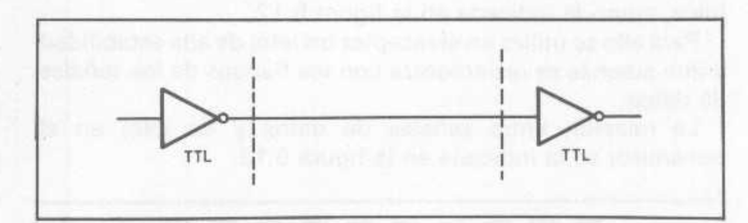


Figura 5.19

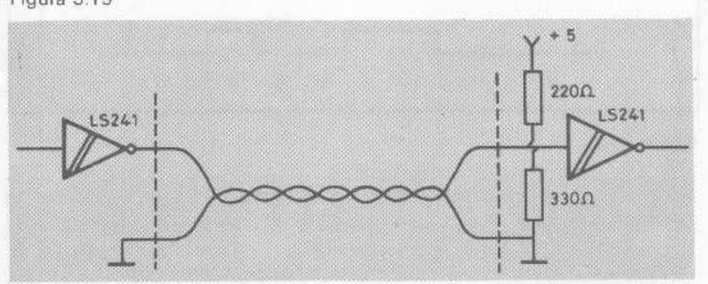


Figura 5.20

— Bucle de corriente. Permite realizar comunicaciones a mayores distancias, hasta 300 metros según la velocidad (normalmente 1200 bps a 30 m y 10 bps a 300 m).

Los niveles 1 y 0 se codifican por la ausencia o presencia de una corriente unidireccional de 20 mA en la línea. Existen las siguientes posibles configuraciones de emisores y receptores para bucle de corriente.

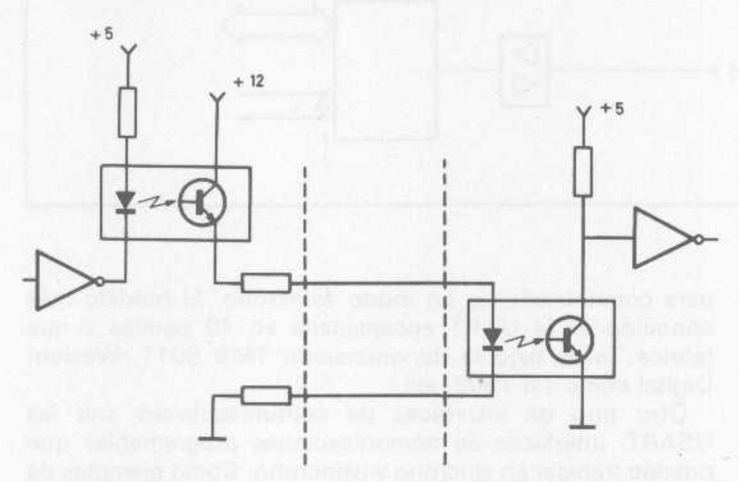


Figura 5.21 Bucle de corriente, emisor activo

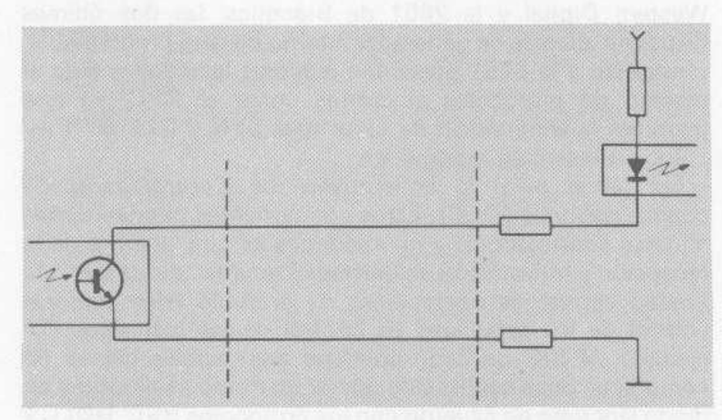


Figura 5.22 Bucle de corriente, receptor activo

— RS-232-C. Utiliza emisores y receptores no balanceados, la señal 1 es una tensión $\leqslant -3$ V y la señal 0 es una tensión $\geqslant +3$ V.

Se utiliza normalmente una señal de +12 y −12 V (la especificación indica ±3 V a ±25 V.).

La velocidad de subida de la señal se limita a 30 V/μs. Esta interfaz está especificada para una velocidad máxima de transmisión de 20 Kbps y una distancia de 15 m.

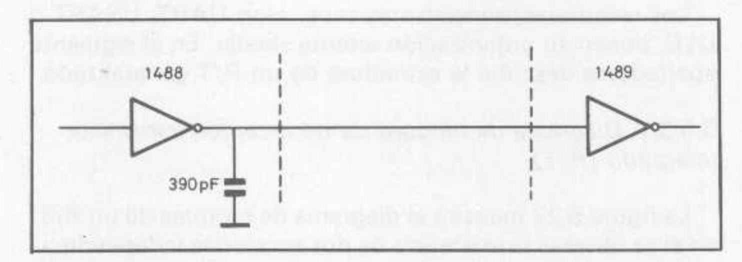


Figura 5.23 Conexión RS-232-C

— RS-422. Utiliza emisores y receptores balanceados (diferenciales) con lo que se consigue una importante mejora en la velocidad y distancia máximas (100 Kbps a 1200 m y 10 Mbps a 12 m).

Se utilizan señales de hasta 6 V y el receptor tiene un umbral de disparo de 200 mV.

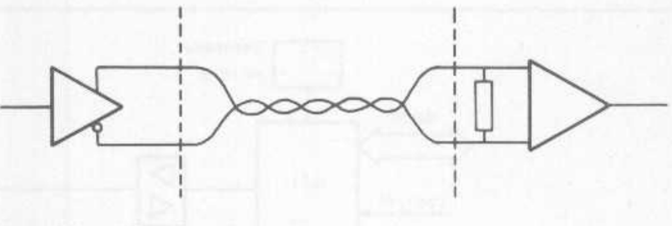


Figura 5.24 RS-422

— RS-423. Utiliza un receptor diferencial y un emisor que no lo es; de esta forma se permite su interconexión con emisores o receptores RS-232C y RS-422.

Las prestaciones que se consiguen son: 300 Kbps a 12 m y 3 Kbps a 1200 m.

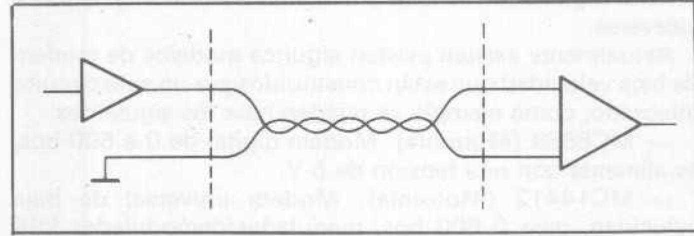


Figura 5.25 Conexión RS-423

5.5 COMPONENTES PARA COMUNICACIONES SERIE

Actualmente el mercado ofrece una serie de circuitos integrados que simplifican la realización de sistemas de entrada/salida serie. Mediante estos componentes se puede construir un sistema de comunicacines serie que desde el lado del procesador ofrece una simple interfaz paralela (fig. 5.26).

El sistema de comunicaciones está compuesto por tres tipos de componentes básicos.

a) Receptores y emisores de línea.
b) Generadores de reloj.
c) Unidades de recepción y transmisión o controladores de comunicaciones.

La misión de cada uno de estos componentes es la siguiente:

5.5.1 Emisores y receptores de línea

Son componentes que realizan la conversión de una señal compatible con circuitos integrados (p. ej. TTL) en una señal adecuada para el manejo de líneas de acuerdo con algún estándar: EIA RS232, RS422, etc. Realizan también la función inversa, conversión de una señal de línea en una compatible con los niveles de los circuitos integrados. Ejemplo de ese tipo de circuitos integrados es el 1488 y 1489 que contienen cuatro conversores de TTL a RS232C y de RS232C a TTL. El 1488 y 1489 es fabricado por diversas compañías (Texas Instruments, Motorola..).

Generalizando un poco el objetivo de los emisores y receptores de línea podrían incluirse aquí los equipos de comunicación de datos o modems. Para comunicaciones entre equipos separados por grandes distancias es útil usar líneas de las compañías telefónicas, utilizando la red conmutada normal o bien alquilando líneas permanentemente conectadas. Tales líneas, diseñadas en principio para comunicaciones de voz, no son adecuadas para la transmisión de señales digitales en forma de impulsos, siendo

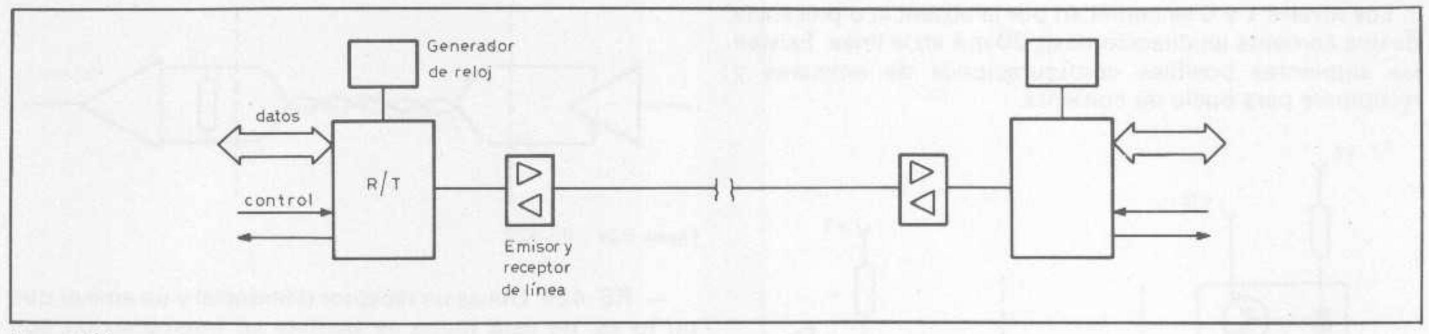


Figura 5.26

precisamente la misión de los modems la conversión de señales digitales en señales analógicas de audiofrecuencia y viceversa.

Actualmente existen existen algunos modelos de modem de baja velocidad que están constituidos por un solo circuito integrado; como ejemplo se pueden citar los siguientes:

— MC6860 (Motorola): Modem digital de 0 a 600 bps; se alimenta con una tensión de 5 V.

— MC14412 (Motorola): Modem universal de baja velocidad, para 0-600 bps, modulador/demodulador FSK compatible con los estándares USA y CCITT para la red de comunicaciones de baja velocidad. Está realizado en CMOS.

5.5.2 Generadores de reloj

Son circuitos que contienen un oscilador, sincronizado generalmente por un cuarzo externo, y una cadena divisora programable para generar una cualquiera de las frecuencias de reloj normalizadas. Por ejemplo, el BR1941 de Western Digital contiene un oscilador y dos cadenas divisoras programables independientemente con 4 bits, pudiendo seleccionar 16 valores de salida distintos: 50, 75, 110, 134,5, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600 y 11200 baudios (la señal de salida es realmente la frecuencia en baudios × 16 para que pueda ser aplicada directamente a una UART). Muchas veces la función del generador de reloj programable está incluida dentro del mismo circuito integrado del receptor y transmisor serie.

5.5.3 Receptores y transmisores serie (R/T)

Los μP pueden ser programados para realizar las operaciones de serialización y deserialización, sincronización de carácter o bloque, generación y detección de paridad y generación y detección de los caracteres de control de errores de los bloques. Si bien éste es el sistema más económico para realizar las operaciones de comunicación serie, presenta una serie de inconvenientes:

— No permite alcanzar velocidades de transmisión superiores a los 4800 baudios.

— Consume todo el tiempo de proceso del microprocesador, impidiéndole la realización de otras tareas.

La utilización de los controladores integrados permite superar los dos anteriores inconvenientes con un costo muy reducido.

Los R/T son los encargados de convertir las informaciones paralelo en serie y viceversa, de acuerdo con algún sistema estándar de codificación.

Presentan por un lado una interfaz paralela clásica y por el otro una línea serie.

Son circuitos integrados LSI que pueden llegar a ser muy complejos. Los componentes más conocidos son las UART, para comunicaciones en modo asíncrono. El modelo más conocido es la UART encapsulada en 40 patillas y que fabrica Texas bajo la denominación TMS 6011, Western Digital como TR 1602, etc.

Otro tipo de interfaces de comunicaciones son las USART, interfaces de comunicaciones programables que pueden trabajar en síncrono y asíncrono. Como ejemplos de USART pueden citarse la 8251 de Intel, la ASTRO de Western Digital y la 2651 de Signetics; las dos últimas disponen además de generador interno de reloj programable, y respecto a la 8251 presentan mayores facilidades para el manejo de protocolos síncronos como el BSC, ya que permiten la eliminación de caracteres SYN y DLE-SYN así como su inserción automática.

Un tercer nivel lo constituyen los *Controladores de comunicaciones* (DLC). Estos componentes pueden denominarse controladores y no interfaces ya que, además de la recepción y transmisión en diversos formatos, descargan a la unidad central de operaciones de protoclo tales como el control de los caracteres de control de los mensajes. Por ejemplo, el SIO de Zilog contiene dos canales dúplex de comunicaciones capaces de operar en modo asíncrono y en modo síncrono de acuerdo con los protocolos BSC, HDLC y SDLC.

En modo HDLC efectúa la inserción y detección automática del cero, la generación y detección del carácter de CRC, la inserción de los caracteres FLAG y el reconocimiento del campo de dirección. Es de destacar que permite trabajar hasta 550 K baudios y que sus registros de recepción permiten una cola de 4 caracteres.

Otros controladores son el 9903 de Texas que maneja también los protocolos BSC, HDLC y SDLC y el 1973 de Western Digital, similar al 8273, es específico para SDLC y HDLC.

Los receptores/transmisores serie, sean UART, USART ó DLC, tienen su organización interna similar. En el siguiente apartado se describe la estructura de un R/T generalizado.

5.5.3.1 *Diagrama de bloques de un receptor/transmisor integrado (R/T)*

La figura 5.27 muestra el diagrama de bloques de un R/T; en él se observa la existencia de dos secciones independientes capaces de operar en paralelo.

Estas dos secciones son: El transmisor y el receptor.

El transmisor está compuesto por:

Registro tampón de transmisión (THR, transmitter holding register). Se utiliza para almacenar caracteres en espera de ser transmitidos. En determinados modelos de R/T el THR puede contener varios caracteres formando una cola de espera. El THR se comunica con el exterior mediante líneas de datos (BDT) y una línea de control (WR) por la que el

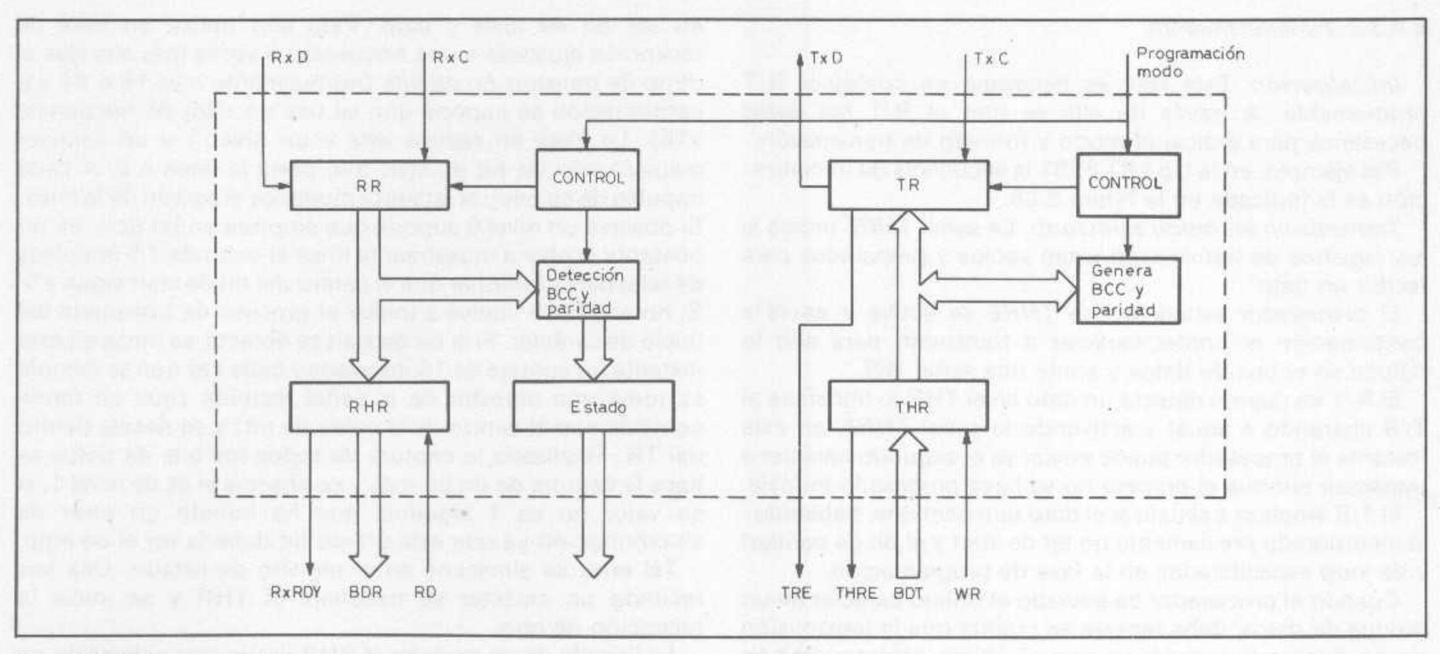


Figura 5.27 Diagrama de bloques R/T

ordenador envía la orden de cargar en el THR el dato de BDT.

Registro de transmisión (TR), encargado de la conversión en serie generando la señal de salida *T×D*.

El TR es un registro de decalaje que utiliza la señal de reloj *T×C*, reloj de transmisión.

Asociados a los registros anteriores existen unas *señales THRE* y *TRE* que indican el estado de los mismos. *THRE* indica que el registro tampón de transmisión está vacío y *TRE* indica que es el registro de transmisión el que lo está.

La señal *THRE* indica al procesador que el THR está dispuesto a aceptar un nuevo dato. El procesador puede entonces colocar el nuevo dato en BD y emitir una señal WR.

La señal *TRE* indica que el registro de transmisión ha quedado vacío. En operación normal en modo síncrono la activación de TE indica que el procesador nos ha enviado a tiempo el siguiente dato, debiéndose considerar tal indicación como error, salvo que el circuito realice la inserción automática de caracteres de relleno.

Circuito de control de transmisión. Este circuito, de acuerdo con señales del exterior, se encarga de reconfigurar la estructura del registro de transmisión para que actúe según el modo y formato de transmisión programado. Así controla si se desean incluir bits de start y stop, el número de bits de stop, el número de bits por palabra, el tipo de bit de paridad a incluir y, en determinados casos, calcula el BCC (carácter de control de errores del bloque) de los datos transmitidos y proporciona el mecanismo para cargar los caracteres BCC en el TR para su transmisión al exterior. La composición de la sección receptora es la siguiente:

Registro de recepción (RR), realiza la conversión de la señal serie recibida por R×D en un dato paralelo.

Registro tampón de recepción (RHR), almacena los datos recibidos que aún no han sido leídos por el procesador. En ciertos modelos de R/T el RHR puede almacenar varios caracteres organizados en una cola FIFO. El registro RHR puede ser leído a través de la salida de datos BDR.

Indicador de estado de recepción, constituido por la señal *R × RDY* que indica la presencia de algún caracter en el RHR.

La señal *R × RDY* es normalmente consultada por el procesador o bien utilizada para generar una interrupción; de cualquier forma el procesador responde a la misma leyendo la línea de datos BDR y emitiendo a continuación una señal *RD* que desactiva R × RDY y permite que RHR pueda recibir un nuevo dato.

Si el RR recibe un nuevo caracter y el RHR está ocupado, porque el procesador no ha enviado aún la señal *RD*, se provoca una condición de error denominada *Overrun* (esta condición de error se graba normalmente en el registro de estado).

Circuito de control de recepción, que interpreta las señales del exterior que definen el modo y formato de recepción y de acuerdo con ellos configura los circuitos de recepción. Por otra parte, realiza el cálculo y comprobación del bit de paridad de cada palabra ó el carácter de control BCC del bloque y genera la palabra de estado con las informaciones de error resultado de la comprobación del BCC o de la paridad y, para el modo asíncrono, el error de bit de stop (framing error), que indica error en la estructura del carácter recibido, ya que después del último bit del carácter no se ha detectado el nivel 1 del bit de stop. Las dos secciones, por estar incluidas dentro del mismo circuito integrado, pueden ser totalmente independientes o bien pueden compartir algún circuito, por ejemplo los elementos de control que definen el modo y formato de trabajo, los circuitos de generación y cálculo de CRC y paridad, etc.

En los circuitos integrados R/T del mercado se observan dos tendencias en su orientación, unos están orientados a su utilización en sistemas basados en un bus y otros no.

Los primeros, por ejemplo los USART 8251 y 6850, concentran toda la comunicación con el procesador en un bus bidireccional por el cual se envían y reciben tanto datos como estados, mandatos y datos de programación del modo de operación.

En cambio, los R/T no orientados a bus mantienen líneas unidireccionales separadas para cada señal. Ejemplos de este tipo de circuitos son los UART estándar de 40 patillas como los 6011, 1602, 2017.

La ventaja de los R/T orientados a bus es la simplificación en el esquema de conexionado y la compacidad del circuito total resultante, pero es prácticamente imposible su utilización en sistemas no orientados a bus.

5.5.3.2 *Funcionamiento*

Inicialización. Esta fase es necesaria en cualquier R/T programable. A través de ella se dan al R/T los datos necesarios para indicar el modo y formato de transmisión.

Por ejemplo, en la USART 8251 la secuencia de inicialización es la indicada en la figura 5.28.

Transmisión en modo asíncrono. La señal *THRE* indica si los registros de transmisión están vacíos y preparados para recibir un dato.

El procesador espera a que *THRE* se active y envía a continuación el primer carácter a transmitir, para ello lo coloca en el bus de datos y emite una señal *WR*.

El R/T en cuanto detecta un dato en el THR lo transfiere al T/R liberando a aquél y activando la señal *THRE*; en este instante el procesador puede enviar ya el segundo carácter a transmitir aunque el primero no se haya procesado todavía.

El T/R empieza a serializar el dato que contiene, habiéndole incorporado previamente un bit de start y el bit de paridad y de stop especificados en la fase de programación.

Cuando el procesador ha enviado el último carácter de un bloque de datos, debe tenerse en cuenta que la transmisión no ha finalizado todavía ya que el último carácter está en THR y el penúltimo se está serializando en el TR. El momento exacto en que finaliza la salida del último carácter por la línea de datos serie lo señala la activación de la señal *TRE*.

Recepción en modo asíncrono. El circuito de recepción realiza las operaciones de sincronización de bits basándose

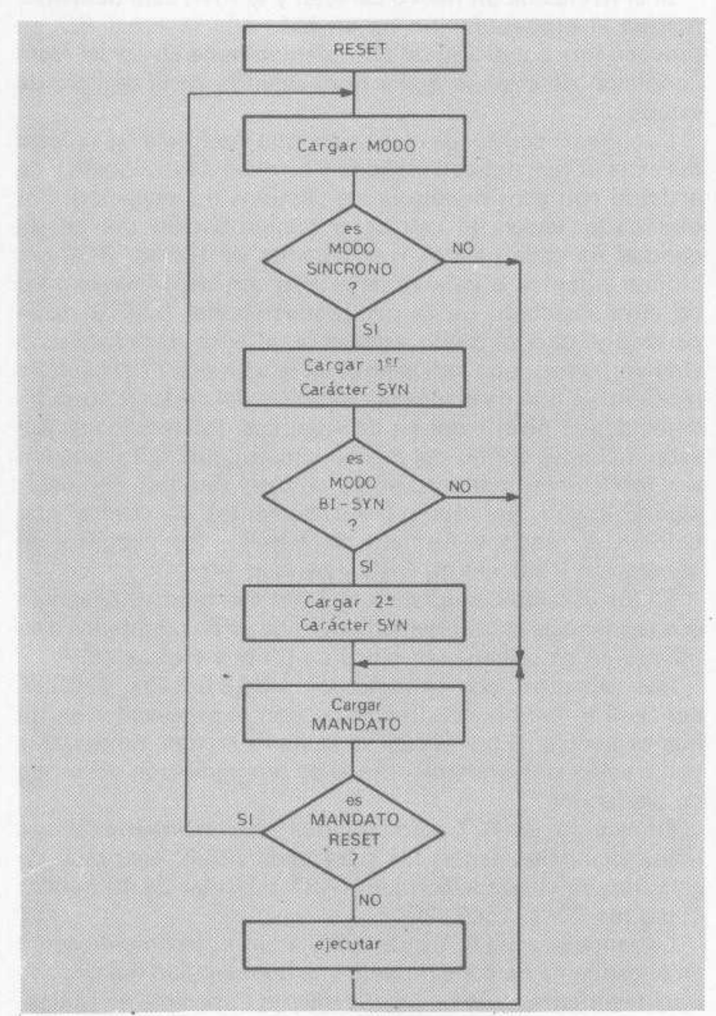


Figura 5.28 Programación de una USART tipo 8251

en los bit de start y stop. Para ello utiliza un reloj de recepción ajustado a una frecuencia *n* veces más alta que el ritmo de transmisión de bits (normalmente *n* es 16 ó 64 y a continuación se supone que se usa un reloj de frecuencia x16). La línea en reposo está a un nivel 1 y un carácter empieza con un bit de start que pone la línea a 0. A cada impulso de su reloj, el receptor muestrea el estado de la línea. Si observa un nivel 0 supone que empieza un bit de start; no obstante vuelve a muestrear la línea al cabo de 7,5 impulsos de reloj para confirmar que el centro del bit de start sigue a 0. Si no es así, se vuelve a iniciar el proceso de búsqueda del inicio de carácter. Si el bit de start se detecta, se inicia en este instante un contaje de 16 impulsos y cada vez que se cumple se toma una muestra de la señal recibida (que en teoría coincide con el centro de la celda de bit) y se decala dentro del TR. Finalizada la captura de todos los bits de datos se hace la captura de un bit más y se observa si es de nivel 1, si su valor no es 1 significa que ha habido un error de sincronización ya que este último bit debería ser el de stop.

Tal error se almacena en el registro de estado. Una vez recibido un carácter se transfiere al THR y se inicia la recepción de otro.

La llegada de un carácter al RHR provoca la activación de la señal $R \times RDY$.

El procesador, en cuento observa que la señal $R \times RDY$ se activa, emite una orden de lectura del registro de recepción enviando una señal *RD* que desactiva $R \times RDY$ y prepara el RHR para la recepción de un nuevo carácter.

Transmisión en modo síncrono. En modo síncrono la transmisión se activa normalmente a través de un mandato o señal de activación, los datos se transfieren del THR al TR en donde se serializan, emitiéndose un bit en cada flanco negativo del reloj $T \times C$.

Una vez iniciada la transmisión, ésta continua interrumpidamente al ritmo del reloj de transmisión. Cada vez que se pasa un dato del THR al TR se activa la señal *THRE*, para que el procesador envie un nuevo dato, si éste no lo hace antes de que finaliza la señalización del último dato, se insertan automáticamente caracteres SYN en la corriente de bits de salida.

Recepción en modo síncrono. Una recepción en modo síncrono se inicia por un mandato específico para que el R/T se ponga en modo búsqueda (hunt mode).

En modo búsqueda, el receptor muestrea en el flanco de subida del reloj $R \times C$ la señal recibida y, a cada tiempo de bit, compara si los últimos bits recibidos coinciden con él o los caracteres SYN especificados en fase de iniciación. En cuanto detecta los caracteres de sincronización pone a cero el contador de bits e inicia la lectura de los datos, que ya están sincronizados a nivel de carácter. Cada carácter recibido se transfiere al RHR y provoca la activación de la señal $R \times RDY$.

Esta señal indica al procesador que puede leer ya el dato recibido y en cuanto lo haga se desactivará la señal $R \times RDY$, en cambio, si no lo hace antes de que se transfiera al RHR el próximo carácter recibido, se activará en la palabra de estado el bit que indica el error de dato perdido (overrun error).

En algunos R/T los caracteres SYN recibidos son automáticamente descartados y no se envían al RHR ni provocan la activación de $R \times RDY$.

5.6 EJEMPLO DE APLICACION

Se presenta como ejemplo de aplicación el diseño de una unidad de comunicaciones asíncronas para un μC.

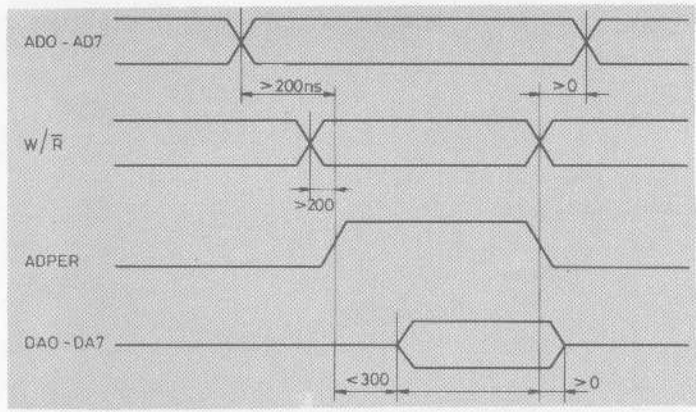


Figura 5.29

Para mayor generalidad se supone que el microcomputador utiliza un bus que cumple el estándar Euromicro.

El bus Euromicro utiliza las siguientes señales para el control de periféricos:

ADO-AD7	Líneas de selección de periférico.
DAO-DA7	Líneas de datos, bus bidireccional.
ADPER	Orden de lectura/escritura en periférico.
ADMEM	Orden de lectura/escritura en memoria.
W/$\overline{R}$	Selector de lectura o escritura (1 escritura, 0 lectura).
RESET	Orden de puesta a 0.

Las especificaciones de tiempo son las mostradas en la figura 5.29.

5.6.1 Descripción del circuito

El circuito compuesto por el decodificador 74155 y las puertas NO-O-exclusiva constituye la unidad de decodificación de órdenes de entrada y salida (fig. 5.30).

Los microinterruptores S1 seleccionan el campo de direcciones a que responderá la UART, cuando los 6 bits de mayor peso de la dirección coincidan con lo programados; las 6 salidas de las puertas NO-O-exclusiva estarán a 1 y si además ADPER está activo, indicando que el bus de direciones contiene la dirección de un periférico, la salida $\overline{SEL}$ se pondrá a 0.

La activación de $\overline{SEL}$ libera a los decodificadores 74155. Una de las 8 salidas se activará; será una de las salidas *E* si W/$\overline{R}$ es 1 o una *L* si W/$\overline{R}$ es 0. La *E* ó *L* en concreto que se activará será la que corresponda según los dos bits bajos de dirección.

Para la generación de los relojes de transmisión y recepción, se ha elegido el circuito integrado BR1941L que es capaz de generar 16 frecuencias de salida seleccionables mediante 4 microinterruptores (S2).

Las frecuencias generadas se corresponden con la tabla 5.1.

El resto del circuito está formado por la UART y los circuitos de adaptación al bus y al exterior.

La interfaz con la exterior utiliza los circuitos integrados 1488 y 1489 que cumplen las especificaciones eléctricas RS232C. Para la conexión de la UART al bus no se ha utilizado la característica «tri-state» de sus salidas porque el bus representa una carga superior a la máxima autorizada por la UART.

Es aconsejable que los circuitos de interfaz con el bus tengan un alto nivel de inmunidad al ruido, entradas de poco consumo y salidas con gran capacidad de control de intensidad. Tales características las cumple el 74LS241, interfaz de bus tri-state con entradas con histéresis.

Las señales *RRD* y *SFD*, que son las encargadas de activar las salidas de los registros de datos recibidos y de estado, se mantienen activas permanentemente y son las puertas tri-state situadas detrás de las líneas de datos las que se controlan mediante los mandatos de lectura $\overline{L0}$ y $\overline{L1}$.

La señal $\overline{L0}$, además de abrir las puertas de datos, es

S2	frecuencia	baudios
0000	0.8	50
0001	1.2	75
0010	1.76	110
0011	2.15	134.5
0100	2.4	150
0101	4.8	300
0110	9.6	600
0111	19.2	1200
1000	28.8	1800
1001	32.0	2000
1010	38.4	2400
1011	57.6	3600
1100	76.8	4800
1101	115.2	7200
1110	153.6	9600
1111	307.2	19200

Tabla 5.1

SBS	0	Un bit de stop.
	1	Dos bits de stop
PI	0	Paridad activada.
	1	Paridad desactivada.
EPE	0	Paridad impar.
	1	Paridad par.
WLS	00	Palabra de 5 bits.
	01	Palabra de 6 bits.
	10	Palabra de 7 bits.
	11	Palabra de 8 bits.

Tabla 5.2

```
THRE    EQ     00000001B   ; máscara para la señal THRE
DR      EQU    00000100B   ; máscara para la señal DR
ERR     EQU    00111000B   ; máscara para las señales de error
DAEN    EQU    00H         ; dirección canal de datos recibidos
DASA    EQU    00H         ; dirección canal de datos transmitidos
STAT    EQU    01H         ; dirección canal de señales de estado
;
EC:     IN     STAT
        ANI    THRE
        JZ     EC
        MOV    A,C
        OUT    DASA
        RET
;
RC:     IN     STAT
        ANI    DR
        JZ     RC
        IN     STAT
        ANI    ERROR
        JZ     BIEN
        MVI    OOFH
        CALL   EC
        JMP    RC
BIEN:   IN     DAEN
        MOV    C,A
        CALL   EC
        RET
```

Tabla 5.3

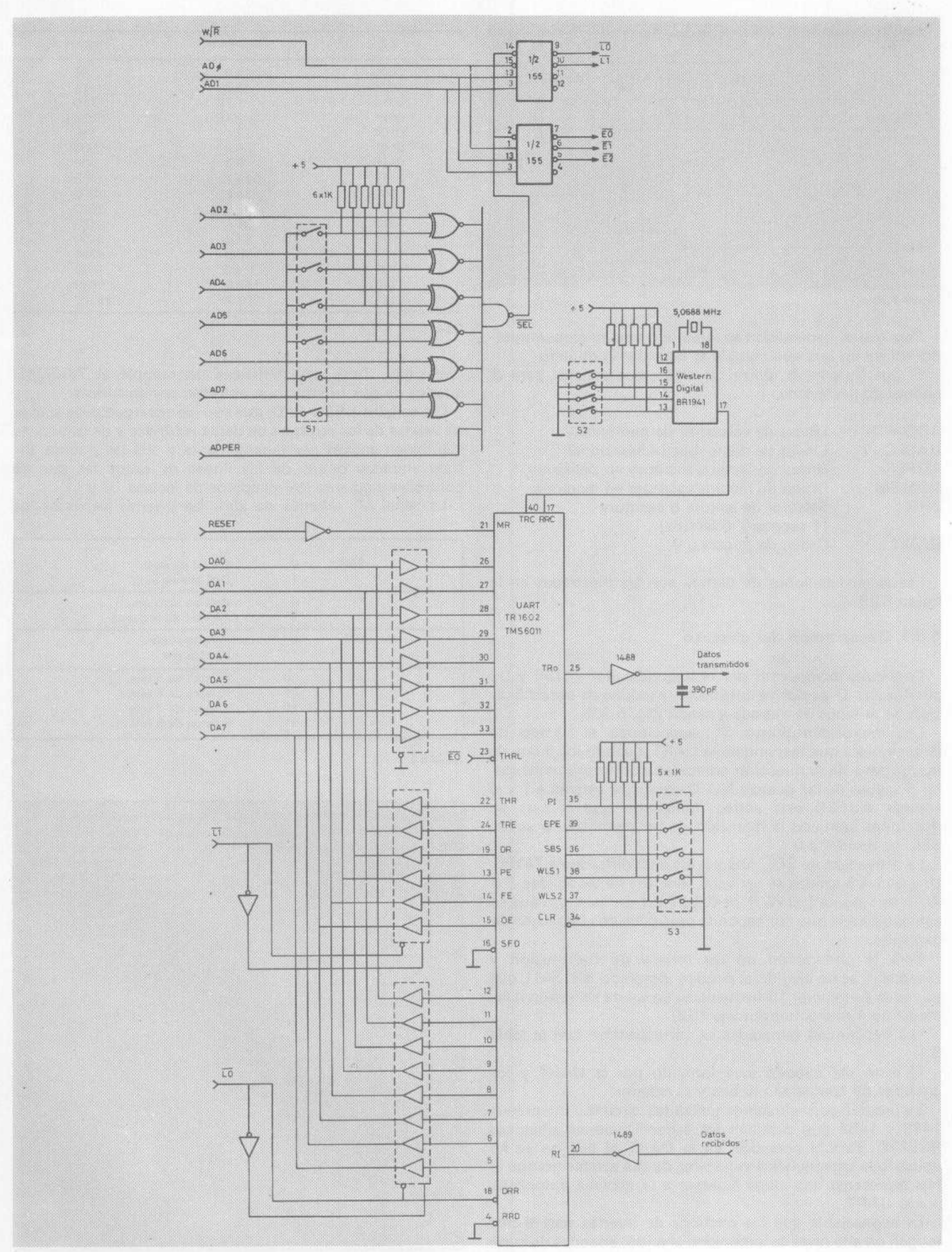


Figura 5.30 Contolador comunicaciones asíncronas para microcomputador

Fabric.	Modelo	Veloc. (bps)	Protocolo B S C	H D L C	S D L C	A D C C P	Tecno-logía	Aliment. (V)	Encapsul. (n.º patillas)	NOTAS
AMD	9551	56 K (sincr.) 9,6 K (asínc.)	—	—	—	—	NMOS	5	28	Compatible con la USART 8251
AMI	6851 PCI	0,8 M	Soft	Soft	Soft	Soft	NMOS	5		Programable por software para cualquier protocolo
Fairchild	F 3846 SPCC	1 M	Sí	Sí	Sí	Sí	NMOS	5	40	Interfaz con bus de 8 ó 16 bits
INTEL	8251	56 K (síncr.) 9,6 K (asíncr.)	—	—	—	—	NMOS	5	28	
INTEL	8273	64 K	—	Sí	Sí	—	NMOS	5	40	
INTERSIL	6101		—	—	—	—	CMOS		40	
INTERSIL	61023	250 K	—	—	—	—	CMOS		40	Versión CMOS de la UART estándar
MOTOROLA	6850 ACIA	500 K	—	—	—	—	NMOS	5	24	UART con interfaz con el modem integrado 6860
MOTOROLA	6852 SSDA	600 K	—	—	—	—	NMOS		24	Interfaz síncrona para diskette, cinta y comunicaciones
MOTOROLA	6854 ADLC		—	Sí	Sí	Sí	NMOS		40	
MOSTEK	3884	500 K	Sí	Sí	Sí	—	NMOS	5	40	Compatible con la SIO
NATIONAL	8250 AC	56 K	—	—	—	—	NMOS	5		Incluye generador de reloj
RCA	18i54 UART	200-400 K	—	—	—	—	CMOS		40	
Rockwell	10930 SDC	250 K	—	—	—	—	PMOS		42	Entradas y salidas RS-232-C
Signetics	2651 PCI	0,8 M	—	—	—	—	NMOS	5	28	
Signetics	2652 MPCC	0,5 M	Sí	Sí	Sí	Sí	NMOS	5	40	
SMC	2502 UART	25-56 K	—	—	—	—	PMOS		40	
SMC	8502 UART	100 K	—	—	—	—	NMOS	5	40	Versión NMOS de la anterior
SMC	2017	25-40 K	—	—	—	—	PMOS		40	
SMC	8017	100 K	—	—	—	—	NMOS	5	40	Versión NMOS de la anterior
SMC	1671 ASTRO	1 M	—	—	—	—	NMOS		40	Compatible en la 1671
SMC	5025 USRT	2 M	Sí	Sí	Sí	Sí	NMOS	5, 12	40	
Synertek	6551	19200	—	—	—	—	NMOS	5	28	
TI	9902 ACC.	250 K	—	—	—	—	NMOS		18	UART con interfaz para CRU del 9900
TI	9903	500 K	Sí	Sí	Sí	—	NMOS	5	20	USART con interfaz para CRU del 9900
TI	6011	200 K	—	—	—	—	NMOS	5,12	40	UART compatible con la 1602
Western Digital	1602	200 K	—	—	—	—	NMOS	5,12	40	
Western Digital	1671.1971 ASTRO	1 M	—	—	—	—	NMOS	5,12,-5	40	
Western Digital	1933 SDLC	1,5 M	—	Sí	Sí	Sí	NMOS	5	40	
Western Digital	1983	20 K	—	—	—	—	NMOS	5	28	Versión de bajo costo de la 8251
Western Digital	1482	640 K	—	—	—	—	NMOS	5,12	40	
ZILOG	SIO	550 K	Sí	Sí	Sí	—	NMOS	5	40	

Tabla 5.4

enviada a DRR para indicar a la UART que ya se ha leído el dato y para que desactive la señal *DR*.

El conjunto de microinterruptores S3 se utiliza para seleccionar el formato de la transmisión que se hace de acuerdo con la tabla 5.2.

Una variante de este esquema podría ser uno en que se sustituyan los microinterruptores S2 y S3 por dos registros de carga en paralelo conectados al bus de datos y que se cargan bajo control de las señales $\overline{E1}$ y $\overline{E2}$. De esta forma, el microprocesador podría programar mediante dos palabras de control tanto la frecuencia como el formato de la transmisión.

Para completar la exposición se adjuntan en la tabla 5.3 las rutinas de tratamiento de esta unidad de comunicaciones. Se utilizará el ensamblador 8080 y se supondrá que la dirección escogida por la placa es 0000 00XX. Se describen dos rutinas básicas:

EC enviar carácter; el dato enviado debe colocarse en el registro C

RC recibir carácter y enviarlo de nuevo como eco si se ha recibido correctamente, o enviarlo con el carácter FFH si se recibe con error; en tal caso se inicia la recepción de un nuevo carácter.

El dato recibido se devuelve en el registro C

5.7 APENDICE

En la tabla 5.4 se muestran las características de los controladores e interfaces de comunicaciones existentes en el mercado.

6 sistemas de control

Santiago Insa Alfaro

6.1 INTRODUCCION

La proliferación de circuitos de control que actualmente se está produciendo, hace difícil la decisión de los diseñadores en el momento de elegir un componente. No obstante, es esta proliferación la que permite un avance y simplificación enormes en los diseños con microprocesadores, extendiendo el uso de estos en campos que parecían hasta hoy no pertenecerles.

Paralelamente, el precio de los conjuntos basados en circuitos altamente integrados ha bajado drásticamente, obteniéndose relaciones potencia/precio muy satisfactorias y que aconsejan a su vez, el uso de tales conjuntos.

Para el diseñador esto supone el que cada día deba emplear más tiempo en la prospección y estudio de los circuitos que más se acomodan a sus necesidades.

A pesar de la gran variedad de circuitos existentes, destacan ciertas tendencias convergentes que cabría extrapolar a los circuitos que en un futuro puedan ir apareciendo. Señalemos a continuación alguno de estos puntos:

1) Prácticamente todo nuevo circuito de alta escala de integración (LSI) o muy alta escala de integración (VLSI), estará diseñado de forma que cumpla algún determinado protocolo de bus ya existente.

2) Se prevé que los nuevos circuitos que aparezcan podrán ser fácilmente integrables en los sistemas de una forma general, es decir, podrán conectarse sin dificultades con cuantos microprocesadores sea posible, sin limitarse únicamente o alguna familia excesivamente concreta.

3) Observando de cerca la aparición de nuevos procesadores integrados del tipo Z8000 y MC68000 de arquitecturas con ambiciones de miniordenador, cabe esperar que aparezca una nueva gama de circuitos integrados de control para potenciar a dichos procesadores, ya que las antiguas familias de circuitos estaban orientadas al manejo de octeto, y resultan inservibles para su interconexión con los citados procesadores de 16 bits.

En este punto cabe citar como excepción los circuitos de control y apoyo de la familia INTEL que pueden ser, en general, conectables con el nuevo microprocesador 8086 de dicha familia. Aunque puede decirse también que el citado 8086 es un microprocesador que podríamos llamar de «puente» entre los de ocho bits y los de dieciséis, sin entrar del todo dentro de este último campo.

4) A medida que aumenta la complejidad y capacidad de los circuitos de control, podrán montarse estructuras más complejas y potentes con menos circuitería.

Gracias a estos circuitos de control podrán montarse arquitecturas que permitan a los procesadores trabajar tanto como sistemas únicos —un solo procesador— como sistemas múltiples jerarquizados maestro-esclavo o sin jerarquizar, pudiéndose con cierta facilidad intercambiar información y tareas entre ellos.

También se podrán ver circuitos de control que permitan no sólo el intercambio de unidades de un mismo bus, sino de buses diferentes entre sí, manejando las prioridades o turnos de interconexión que se establezcan.

5) Cabe esperar que segundas y hasta terceras generaciones de circuitos compatibles con sus predecesores —compatibilidad al nivel de patilla— incorporen en éstos características mejoradas así como correcciones de las deficiencias de funcionamiento de los viejos circuitos anteriores, todo esto como resultado de la experiencia de uso de dichos circuitos.

6) Puede observarse también el incremento de acuerdos entre casas fabricantes para ser segundas y terceras fuentes de unos mismos circuitos, así como para cooperar en el diseño de nuevos controladores, estandarizando en cierta medida los elementos y asegurando un suministro a precios más bajos que los que presenta un fabricante único proveedor de un determinado producto.

Todas estas espectativas están pasando día a día de promesas a realizaciones, abriendo cada vez mayores posibilidades al diseñador que debe siempre estar pendiente de los adelantos que puedan interesarle.

Entre los tipos de circuitos integrados de control podríamos establecer una primera clasificación según su funcionalidad de la forma siguiente:

- Controladores de interrupciones
- Controladores de acceso directo a memoria
- Controladores de memoria
- Controladores de periferia y basados en microprocesadores

Aunque éstas sean las funciones típicas que actualmente cubren los circuitos del mercado, existen otros no catalogados aquí por realizar trabajos muy específicos y menos generalizables.

Pasemos a analizar y desarrollar cada uno de dichos tipos de controlador, introduciéndonos en los aspectos generales y describiendo alguno de los modelos ya existentes.

6.2 CONTROLADORES DE INTERRUPCIONES

6.2.1 Métodos de trabajo en los sistemas basados en μP

En los sistemas basados en microprocesadores, existen periféricos tales como teclados, visualizadores, sensores, etc. que deben ser atendidos por la unidad de proceso para poder efectuar sus transferencias de datos hacia o desde el sistema, sin cargar excesivamente al procesador con la ejecución de dichas tareas.

Como quiera que las necesidades de atención que estos periféricos requieren suelen surgir de forma aleatoria en el tiempo, la unidad de proceso deberá vigilar constantemente a dichos dispositivos consultando su estado para atenderlos, o bien serán éstos los que, a través de una interrupción,

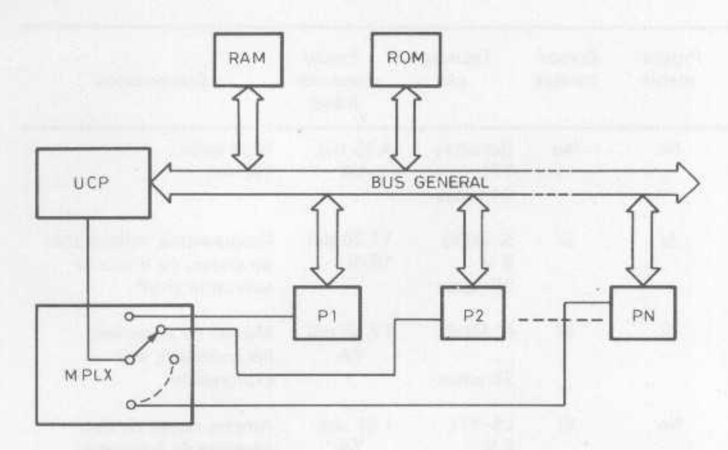


Figura 6.1 Método de exploración. UCP = unidad central de proceso; P1 = periférico 1; MPLX = multiplexor

alerten al microprocesador para que éste les suministre el servicio requerido.

Estas dos diferentes formas de atención a la periferia dan nombre a los dos principales métodos de trabajo en los sistemas basados en microprocesadores. Los métodos son: de barrido o exploración (polled method) y de interrupciones.

6.2.2 Método de exploración

Este método (fig. 6.1) consiste en que el procesador gobierne a un multiplexor el cual hace que cada periférico sea interrogado a través del bus general para que indique al procesador si tiene pendiente o no alguna solicitud de manejo de datos. Si la respuesta es negativa, la unidad de proceso normalmente incrementa en uno la posición del multiplexor y se pasa a preguntar al siguiente periférico. Si la respuesta ha sido afirmativa, se pone en marcha la ejecución del proceso correspondiente a la atención del periférico.

La exploración suele ser cíclica y las prioridades son idénticas para todos los periféricos.

Este método es fácil de realizar pero requiere que el procesador pierda parte de su tiempo consultando a los periféricos, en detrimento de la capacidad de ejecutar procesos de usuario y por tanto con pérdida de rendimiento.

6.2.3 Método de interrupción

Para poder implantar este método (fig. 6.2), la unidad de proceso dispone de una o más entradas de interrupción, que al ser activadas, detienen la ejecución del programa en curso

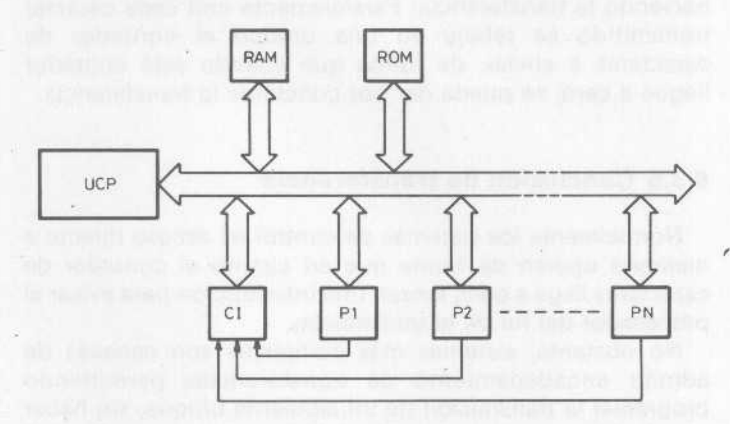


Figura 6.2 Método de interrupción. CI = controlador de interrupciones; P1 = periférico 1

y obligan al procesador a bifurcar hacia la ejecución del programa de atención de periférico. Una vez dicha atención ha concluido, el procesador seguirá la ejecución del programa interrumpido a partir de donde quedó.

Resulta evidente que con este método se incrementa la eficacia del sistema, ya que el microprocesador no se aparta de la ejecución del programa principal más que cuando aparece una necesidad de atención por parte de algún periférico.

Los sistemas de control de interrupciones son los que facilitan al procesador el funcionamiento en modo interrupción. Los más complejos aceptan solicitudes de interrupción de todos los periféricos, determinan cuál de entre las citadas solicitudes es la más prioritaria, y si dicha interrupción es más o menos prioritaria que la posible interrupción que está siendo atendida por la unidad de proceso, y por fin interrumpen a dicha unidad a través de una sola línea.

Por otro lado, los controladores de interrupciones pueden suministrar también la dirección de inicio en donde se encuentra el programa de atención del periférico que acaba de interrumpir, facilitando así la labor de la unidad de proceso que ya no debe preguntar por dicha dirección al periférico.

No todos los controladores de interrupción ofrecen toda la gama de características aquí citadas. La tabla 6.1 muestra una visión general de los circuitos existentes y de algunas de sus características más destacables.

En el apartado 6.6 se describe uno de dichos circuitos, concretamente el 8259 de Intel que es uno de los que presenta una gama más completa de características.

6.3 SISTEMA DE CONTROL DE ACCESO DIRECTO A MEMORIA (ADM)

6.3.1 Generalidades

Estos sistemas de control son los encargados de controlar las transferencias con memoria de la información procedente o que se dirige a periféricos, consistentes básicamente en memorias externas, líneas de comunicación o cualquier dispositivo que presente la necesidad de transmitir bloques grandes de información a la memoria.

En el acceso directo a memoria, el microprocesador será el encargado de programar las transferencias que se necesite efectuar y éstas ocurrirán de forma automática supervisadas por el sistema de control que al final de la operación avisará al microprocesador, bien con una interrupción, bien con un estatus que indique el fin de transmisión, así como las incidencias habidas durante dicha transmisión.

Pasemos a describir las funciones que un sistema de control de acceso directo a memoria debe llevar a cabo, con las posibles variantes que son usuales en sistemas basados en microprocesadores. El estudio de dichas funciones lo realizaremos siguiendo el hilo de una transferencia a memoria.

6.3.2 Programación

En primer lugar, estos sistemas de control deberán contener los registros necesarios para su programación. En dicha programación el microprocesador con instrucciones de entrada-salida, deberá indicar al contador la dirección exacta de memoria en donde deba hacerse la primera transferencia, así como el número de caracteres a transmitir. También puede programarse la información relativa al

Compatible con bus de	Suministrador	Modelo	Velocidad	Nivel de prioridad	Ampliable	Programable	Enmascarable	Tecnología	Precio disponibilidad.	Comentarios
8080	Intel	8214 PICU	2 MHz 3 MHz	8	Sí	No	No	Schottky TTL 24 patas	4,65 dól. YA	Muy veloz (50 ns)
8080	Intel	8529	2 MHz 3 MHz	8	Sí hasta 64 niveles	Sí	Sí	N-MOS 5 V 28 patas	17,20 dól. 1979	Programable, multimodo, da direcc. de inicio de subrutina al μP.
General	AMD	9519	2 MHz 4 MHz	8	Sí	Sí	Sí	N-MOS 28 patas	17,50 dól. YA	Modos de prioridad: fija, rotatoria, por exploración
9900	TI	9907	2-4 MHz	8	Sí	No	Sí	LS-TTL 5 V 16 patas	1,01 dól. YA	Amplio rango de temperatura de funcionamiento (74LS148)
9900	TI	9909	2-4 MHz	8	Sí	No	Sí	LS-TTL	1,22 dól. YA	Amplio rango de temperatura de funcionamiento (74LS348-tres-Estado)
9900	TI	9961	2-4 MHz	16	No	Sí	Sí	I^2L 40 patas	? 1979	Incorpora temporizador de intervalos en el mismo circuito
6800	Motorola	MC6828		8		Sí	Sí	TTL	YA	

Tabla 6.1 Controladores de interrupciones

periférico indicando el lugar exacto en donde debe ser ubicado o extraída la información, aunque estos datos pueden enviarse directamente al controlador del periférico en vez de al controlador de acceso directo a memoria.

En algunos sistemas más complejos se programa solamente la dirección de inicio de memoria y de allí se extrae el resto de información necesaria para la programación de la transferencia. Esto simplifica las entradas y salidas necesarias para programar al controlador.

También se suele especificar, si el sistema de control lo permite, la modalidad de acceso a memoria que se va a efectuar. A medida que se avance en la descripción general veremos algunas de estas modalidades.

6.3.3 Solicitud

Cuando el controlador posee todos los datos relativos a la transferencia que se ha programado, solicitará el envío del primer carácter. Para ello deberá poder usar el bus de acceso a memoria y la memoria propiamente dicha. En los sistemas basados en microprocesadores suele haber un único bus por el que se unen todos los subconjuntos del sistema, y normalmente el uso de dichos buses está gestionado por el microprocesador de tal manera que solicitar el uso del bus para acceder a memoria consiste en la activación de una señal que trasciende al microprocesador. Si el sistema no se ha montado *Unibus* se debe solicitar el uso del bus de memoria, al controlador de dicho bus, que es el que debe dar la oportuna autorización de transferencia.

El sistema de gestión del bus —normalmente el microprocesador— contestará con la adjudicación correspondiente del bus. Para ello previamente habrá priorizado convenientemente la orden de adjudicación, pudiendo existir un sistema muy variado de prioridades para tal fin. Lo normal es que se usen sistemas sencillos de prioridad como es el de asignación fija según el orden físico de ubicación en el sistema.

6.3.4 Transferencia

La adjudicación del bus puede ser solo para un carácter o bien para todo el bloque. Es decir, una de las modalidades que puede haber sido programada en el controlador es que éste, una vez comenzado la transferencia, retenga el bus hasta concluir todo el bloque de información y ceda el bus al concluir completamente ésta. Lo usual es que la transferencia se haga carácter a carácter, liberando el bus en cada uno de ellos a fin de que puedan intercalarse extracciones de instrucciones por parte del procesador, o bien que pueda entrelazarse otro periférico haciendo a su vez accesos a memoria.

En cada carácter, la dirección de memoria será incrementada por el controlador para apuntar a la dirección siguiente. Aquí también cabe una variante de modalidad de transferencia consistente en decrementar la dirección de memoria correspondiente al siguiente carácter.

El controlador solicita todos los accesos lanzando ciclos de memoria a petición del periférico con el que está haciendo la transferencia. Paralelamente con cada carácter transmitido se rebaja en una unidad el contador de caracteres a enviar, de forma que cuando este contador llegue a cero, se pueda dar por concluida la transferencia.

6.3.5 Conclusión de transferencia

Normalmente los sistemas de control de acceso directo a memoria operan de forma que en cuanto el contador de caracteres llega a cero, lanzan una interrupción para avisar al procesador del fin de la transmisión.

No obstante, sistemas más complejos son capaces de admitir encadenamiento de transferencias permitiendo programar la transmisión de un siguiente bloque, sin haber concluido con la transferencia del bloque en curso. De esta forma el procesador puede encadenar transferencias hacien-

do o no que interrumpa con el final de cada una de ellas. La modalidad de encadenamiento o no, y la de emisión de interrupción o no al final del bloque deben programarse al principio antes de la orden de transferencia del bloque.

Un paso más en complejidad es el funcionamiento en modo «canal» aunque este sistema ya no es muy usual en conjuntos basados en microprocesador. En esta modalidad el controlador interpreta además los últimos caracteres transferidos de memoria, como mandatos que debe ejecutar. En ellos se le puede ordenar al controlador que encadene una nueva transferencia de bloque, indicándole la posición de memoria en donde se encuentran los datos de la transferencia —número de caracteres a transmitir, posición inicial, modalidad, etc.— o bien se le puede ordenar que interrumpa al procesador al final del bloque. De todas las maneras, estos sistemas tan inteligentes no son muy corrientes en equipos relativamente pequeños.

La gama de circuitos controladores que pueden encontrarse en el mercado es amplia. La tabla 6.2 pretende dar una visión de los tipos más importantes. Muchos de ellos permiten conectar más de un periférico multiplicando así el número de registros internos que contienen. Los ciclos de acceso a buses de memoria se realizan admitiendo demandas de ciclo de los periféricos —o de los controladores— conectados. Esta demanda de ciclo se arbitra en cuanto a prioridad y el controlador de acceso directo a memoria transfiere el carácter correspondiente al periférico más prioritario, usando los registros contadores, de dirección, etc., asignados a éste. El apartado 6.7 describe un controlador típico (DMAC-MC6844 de Motorola) que refleja el funcionamiento de estos circuitos.

6.4 SISTEMAS DE CONTROL DE MEMORIA

6.4.1 Generalidades

Existe una gran diversidad de tipos de memorias en los sistemas basados en microprocesadores y por tanto, los sistemas de control no son únicos, abarcando desde algunos extremadamente simples hasta otros de gran complejidad.

Normalmente se suele organizar una única memoria principal para todo el sistema, siendo la longitud de palabra del mismo número de bits que el de una instrucción del microprocesador utilizado. Así, si dicho microprocesador al hacer la extracción de una instrucción contenida en una posición determinada espera encontrar un octeto, lo más normal será organizar la memoria por octetos. El controlador en cada ciclo hará la extracción del octeto direccionado.

No obstante, la memoria puede estar distribuida por bancos, de direcciones diferentes, debiendo en dicho caso el controlador saber distinguir si está siendo aludido o no según sea la dirección puesta en el bus de acceso. Esto ya exige un mayor grado de complejidad en el controlador.

Las funciones que en general debe poder llevar a cabo un controlador de memoria son básicamente dos: ejecución de ciclos de lectura y ciclos de escritura. Este sería el caso de un controlador de memorias estáticas, por ejemplo.

Refiriéndonos a memorias de semiconductor —volátiles—, podemos distinguir dos tipos principales: las memorias estáticas y las dinámicas. Las primeras tienen como célula memorizadora un circuito báscula (flip-flop) que conduciendo de un lado o de otro memoriza la información correspondiente por tiempo indefinido, siempre y cuando no se desconecte la tensión de alimentación. Las del segundo tipo basan su célula memorizadora en la carga de un condensador que retiene la información siempre y cuando no vaya descargándose a lo largo del tiempo por las fugas propias del dieléctrico. Como este último fenómeno siempre ocurre, estas memorias deben ser refrescadas —recargadas con su propia información— periódicamente, siendo normalmente la periodicidad de dos milisegundos.

En caso de utilizar memorias dinámicas, el sistema de control deberá generar periódicamente ciclos de refresco complicándose un poco más el diseño.

Suele ser un criterio económico y de fiabilidad el usado para decidirse sobre el uso de un tipo u otro de memorias. Las memorias estáticas acostumbran a tener un precio por bit mayor que las dinámicas, así como menor escala de integración aunque tienen en general buenos tiempos de acceso, son de manejo simple y exigen un controlador menos complicado. Si el tamaño de memoria necesario no es muy grande, suele ser recomendable la utilización de las del tipo estático. Las dinámicas se usan en sistemas mayores en donde su utilización puede proporcionar volúmenes y precios inferiores.

La fiabilidad intrínseca es mayor en las memorias estáticas y menor en las dinámicas, lo cual en sistemas basados en estas últimas obliga a la utilización de redundancias en los bits de datos cuando se desea conseguir un grado alto de fiabilidad. Así, por ejemplo, si la palabra es de ocho bits y se pretende corregir fallos de un bit y detectar errores en dos, pueden utilizarse cinco bits de redundancia según un *código Hamming*, y si la palabra es de dieciséis bits bastarán seis bits extras de código Hamming para poder detectar fallos de dos bits y corregir la información en caso de fallo de uno. Esto exige aumentar el número de circuitos de memoria en un tanto por ciento bastante considerable

Compatible con bus de	Suministrador	Modelo	Velocidad	Canales	Modos	Tecnología	Disponibilidad	Comentarios
8080	Intel	8257	2 MHz 3 MHz	4	Lectura escritura, verificación	N-MOS 5 V 40 patas	Ya	2.as fuentes: National, NEC
8080	AMD	9517	3-4 MHz 2M palabras seg	4	Bloque Demanda 1 palabra Cascada	N-MOS 5 V 40 patas	Ya	2.as fuentes: INTEL 8237,-2
Z-80	Zilog	Z-80 DMA	2,5 MHz 4 MHz	—	—	NMOS 5 V 40 patas	Ya	2.as fuentes: Mostek (3883)
6800	Motorola	6844	1 Mbyte seg.	4	Parado-bloque parado-robociclo robociclo	N-MOS 5 V 40 patas	Ya	2.as fuentes: AMI (68044),
6800	Motorola	68.000 DMA	—	—	—	64 patas	Ya	2.a fuente: Rockwell
9900	TI	TMS 9911	1M palabra seg.	2	Encadenado	LS-TTL V 40 patas	Ya	Ampliable a más canales programable

Tabla 6.2 Controladores de DMA

aunque las tasas de fiabilidad que se alcanzan son muy satisfactorias.

Un sistema más simple que el citado llevaría un único bit de paridad para detectar fallos en los datos aunque careciendo de la posibilidad de reconstruir la información.

En el caso de llevar circuitos de autocorrección por bits de redundancia, el controlador puede dar un aviso —interrupción— al microprocesador comunicándole la corrección del dato y registrando la dirección de error y el código Hamming leído. Si el controlador es más inteligente, él mismo generará un ciclo de escritura con el dato correcto. Si la estadística de fallos supera un cierto valor, puede afirmarse que existe alguna memoria estropeada y si la frecuencia de fallos no es muy elevada puede pensarse en pérdidas aleatorias de bits que entran dentro de las características de fiabilidad dadas por el fabricante.

De lo dicho anteriormente se desprende que en un sistema que al menos detecta fallos en la información debe haber una conexión a la unidad central como periférico de entrada-salida para poder generar interrupciones al microprocesador y para que éste pueda leer la información del estado de la memoria, en donde ésta indicará los detalles del error detectado u otros posibles estados. Tendrá pues unos registros asignados para que en los ciclos de entrada-salida sepa darse por aludida cuando se programa una instrucción de dicho tipo que le haga referencia.

Otros mecanismos que pueden estar incluidos en los controladores son muy variados. Por ejemplo, en los sistemas con más de un banco de memoria, cada uno de dichos bancos debe saber responder cuando la dirección de la palabra a leer o escribir le pertenezca, es decir, esté comprendida entre unos límites que pueden ser fijos o programables. En este último caso deberá aceptar mandatos de cambio de dirección asignada —cambio de límites— pasando después a funcionar con un nuevo conjunto de direcciones. Este sistema se usa a veces cuando se quiere sustituir dinámicamente un banco por otro a efectos de revisión o reparación.

Otro mecanismo que puede hallarse presente en un controlador es el de protección, previa división por páginas, de la memoria. Cuando la memoria es accedida, se consultan los bits de protección asociados a esa posición o a esa página y el controlador actúa en consecuencia. Así, si estuviese protegida de escritura, no llevaría a cabo dicha escritura, por ejemplo.

El mapeado es otro circuito que puede estar incluido en un sistema de control de memorias. El mecanismo del mapeado puede funcionar como un reubicador de las páginas lógicas (conjunto de direcciones que se envían al controlador) en páginas físicas (conjuntos de direcciones reales de memoria física) permitiendo una mejor gestión de la memoria y una ampliación de la memoria física total disponible. De llevar incluido un sistema de mapeado, el controlador deberá disponer de las tablas de reubicación o correspondencia de páginas para hacer en cada acceso la transformación de direcciones correspondiente.

También pueden existir controladores que sepan ejecutar ciclos de lectura-modificación-escritura (en la misma dirección) a fin de simular posiciones autoincrementantes en memoria, autodecrementantes, posibilidad de ejecución de instrucciones de sincronización encendiendo semáforos (bits) en posiciones que sean accedidas, etc. De todas formas, estos mecanismos no son muy usuales en los sistemas con microprocesadores, y entran ya en el terreno de los miniordenadores potentes. No obstante, cada día más, se está elevando la línea divisoria entre estos tipos de sistemas y se están haciendo controladores de memoria más complejos destinados a sistemas basados en microprocesador.

Veamos ahora un poco más detalladamente los ciclos más usuales de los controladores. Los ciclos básicos son el de refresco, el de escritura y el de lectura.

6.4.2 Ciclo de refresco

Este ciclo solo deberá efectuarse si las memorias usadas son dinámicas. Normalmente estas memorias se presentan en circuitos de 4K×1 bit, 16 K×1 bit y 64 K×1 bit, constando de una matriz de células de memoria dispuestas en *N* filas y *N* columnas. Cada vez que se realiza un refresco en una posición queda refrescada toda la fila correspondiente a dicha posición, debiéndose pues refrescar periódicamente las *N* filas y no las *N* columnas. Los fabricantes con la tecnología actual aconsejan refrescar cada célula cada dos milisegundos, pudiendo hacerse un refresco múltiple de *N* ciclos de una vez cada dos milisegundos o repartir en este tiempo los ciclos de refresco, refrescando una fila cada $2/N$ ms. Esta última forma es la más usada.

La pérdida de información en una posición de memoria se debe a las fugas que hacen desvanecer la carga existente. Estas fugas aumentan con la temperatura y se considera el período de 2 milisegundos cuando la temperatura es de 75° C. Si la temperatura es inferior, el tiempo entre refrescos puede aumentar y algunos tratados sugieren el uso de un oscilador con una constante de tiempo *RC* variable (como *R* usan una NTC) para solicitar ciclos de refresco con una frecuencia menor si la temperatura es baja. Lo que si es necesario es disponer de un oscilador que sepa solicitar al controlador un ciclo de refresco cada vez que corresponda.

Por otra parte deberá existir un contador que indique la fila en la que se va a efectuar dicho refresco, es decir con la dirección de fila correspondiente. Este contador será cíclico y su tope será igual al número total de filas a refrescar.

Este oscilador puede ser asíncrono con los ciclos de lectura y escritura de memoria, debiendo existir en el controlador un circuito que efectúe el arbitraje entre solicitudes de ciclo de refresco y solicitudes de ciclos provinentes de bus de acceso (del microprocesador o de los accesos directos a memoria). Normalmente, cuando el controlador está efectuando un ciclo cualquiera (está ocupado) no atiende ningún ciclo hasta que no concluye el que está realizando en esos momentos. Una vez lo acaba, pasa a realizar el ciclo siguiente solicitado, dando prioridad a la petición de refresco sobre otras peticiones.

Otra forma de realizar los refrescos es aprovechar los tiempos muertos entre accesos de lectura y escritura. Así, si por ejemplo, entre accesos del microprocesador o de periféricos de acceso a memoria queda tiempo suficiente para efectuar ciclos de refresco, estos se efectúan aunque se estén haciendo con una frecuencia superior a la requerida por el fabricante de memorias. De esta forma el controlador efectuará ciclos sucesivos de refresco como tarea de fondo y dejará de hacerlos cuando se le solicite por el bus de acceso ciclos de otro tipo. En esta modalidad ya no son prioritarios dichos ciclos de refresco.

6.4.3 Ciclos de escritura

Estos ciclos se efectúan normalmente por solicitud externa a la memoria. Al controlador se le suministra una dirección, unos datos y la función de escribir y a veces también una orden de validación de todo lo anterior, y se espera que el sistema de control devuelva una indicación de escritura

concluida, para que el microprocesador o el periférico continúen con sus ciclos.

Durante el ciclo el controlador deberá temporizar las señales necesarias para las memorias que lleva conectadas a fin de efectuar la escritura correspondiente. En primer lugar puede ocurrir que las direcciones entren multiplexadas en las memorias. Así por ejemplo, en el modelo estándar de 16 K×1 bit se deben enviar al circuito —que está organizado en 128 filas×128 columnas— los siete bits de dirección de fila en un primer envío y los siete bits de columna en un segundo envío, seleccionando además el conjunto de circuitos de memoria que van a recibir las órdenes de escritura enviando solamente a estos circuitos los impulsos necesarios.

La manipulación de los datos a escribir puede ser inexistente reduciéndose únicamente a dar más potencia de salida o bien puede complicarse con la generación de un bit de paridad o de un conjunto de bits de redundancia según un código determinado, a fin de almacenar esta información complementaria al dato escrito.

Por último, el controlador genera la señal de fin de ciclo necesaria para que sea recibida por el dispositivo que solicitó el ciclo.

6.4.4 Ciclos de lectura

Al controlador se le deberá indicar la dirección de memoria acompañada de la función «leer» así como la orden de validación que puede ser la propia función de lectura.

La manipulación que el controlador lleva a cabo con las direcciones es análoga a la que se efectúa en el ciclo de escritura.

En cambio, en los datos se deberá seguir un proceso inverso. Así, si en la escritura se generó la paridad o un código corrector, en la lectura se deberá comprobar si el bit de paridad o bits de código leídos son correctos o acordes con el dato. Si todo está en orden, el controlador pondrá dichos datos en el bus para que sean recogidos por el solicitante y generará una señal que valide dichos datos y concluya el ciclo.

Si hubiese existido una anomalía en los datos, el controlador, si puede, corregirá el dato leído. Lo corregirá interrumpiendo al procesador e indicando en los registros de estado los datos relativos al fallo. Si no dispone de bits de corrección y sólo dispone de bit de paridad, también interrumpirá para que, si el resto de la memoria no falla, el procesador pueda poner en marcha la rutina correspondiente para manejar dicha interrupción.

Si el controlador es de modelo más avanzado intentará recomponer la información, autogenerándose una escritura en memoria con el dato correcto, aunque esto ya no es muy usual.

En la tabla 6.3 se pretende resumir los circuitos integrados existentes actualmente y que sirven como controladores de memoria.

En el apartado 6.8 se examina brevemente un circuito de control de memorias dinámicas, su esquema interno y las funciones que realiza. En concreto se trata del circuito 8202 de la casa INTEL.

6.5 CONTROLADORES BASADOS EN MICROPROCESADORES

6.5.1 Introducción

La enorme difusión de los microprocesadores ha hecho

Suministrador	Modelo	Tecnología	Disponibilidad	Función
Intel	8206	68 patas S-TTL	Ya	Detector/corrector error (chip cuadrado)
Intel	8202	40 patas	Ya	Controlador memorias dinámicas 16 K × 1 hasta 64 Kbytes
Intel	8203	S-TTL 40 patas	Ya	Controlador memorias dinámicas 64 K × 1 hasta 256 Kbytes
Intel	8207			Controlador memorias dinámicas doble puerta acceso
AMD	2960	48 patas	Ya	Detector/corrector de error
AMD	2961	LS-TTL 24 patas	Ya	Buffer de datos de múltiple acceso
AMD	2964	LS-TTL 40 patas	Ya	Controlador memorias dinámicas 64 K × 1
AMD	2965/6	LS-TTL 20 patas	Ya	Drivers de 8 líneas de direcciones de memoria
Fairchild	9449	I^3L 64 patas	Muestras	Gestor memoria
Fairchild	9446	I^3L 64 patas	Muestras	Controlador de memoria
Fujitsu	MB 1412A	ECL/TTL 64 patas	Ya	Detector/corrector de error (chip cuadrado)
Motorola	MC6829	N-MOS 40 patas	Ya	Unidad de mapa de memoria
Motorola	MC3480/ 3232/3242	TTL 24/28 patas	Ya	Controlador de memoria
Motorola	MC68451	N-MOS 64 patas	Ya	Unidad de manejo de memoria para el μP 68.000
Motorola	MC68450	S-TTL 48 patas	Muestras	Detector/corrector de error
National	NS16082	N-MOS 24/28 patas	Ya	Gestor memoria
TI	74LS600/ 630/631	LS-TTL 28 patas	Ya	Detector/corrector de error
TI	74LS610/ 11/12/13	LS-TTL 40 patas	Ya	Unidad de mapa de memoria
Zilog	Z8010	N-MOS 48 patas	Ya	Gestor memoria

Tabla 6.3 Controladores de memoria

posible su aplicación en multitud de campos diversos que van desde complejos sistemas informáticos, hasta sustitutivos de placas de electrónica de funciones muy limitadas.

Puede decirse que la amplia gama de microprocesadores existentes en el mercado cubre completamente las necesidades de un sinúmero de aplicaciones, que si bien podrían resolverse con circuitos lógicos convencionales, sería antieconómico o impracticable realizarlos de esa manera.

En el mercado se pueden encontrar microprocesadores muy completos incluso de 16 bits, muy baratos y con baja velocidad. Si la velocidad es necesaria, entonces puede ascenderse en la gama existente, pasar de la tecnología MOS a TTL Schottky, en la que con detrimento de la integración se pueden cubrir los requerimientos de velocidad que se precisan.

Más adelante se verán con mayor detalle los criterios que permiten seleccionar el microprocesador que mejor se adapta a las necesidades del diseño.

6.5.2 Aplicaciones

Las aplicaciones de los microprocesadores usados como controladores se pueden agrupar en niveles funcionales. Así, según el uso a que se destinen se encuentran las siguientes categorías.

1) Utilizados como sustitutivos de servomecanismos de control y mando
2) Usados para controlar pequeñas unidades autónomas
3) Usados para el control de periféricos en sistemas de micro o miniordenador

No obstante, a veces resulta difícil encuadrar un sistema concreto en alguno de estos tres niveles, aunque esta clasificación resulta útil para describir conceptualmente estas áreas o campos de aplicación.

En el nivel 1 se intentan encuadrar todas aquellas aplicaciones que hacen aparecer al microprocesador como un sistema realimentado en donde unos valores de consigna se combinan con los valores medidos en el sistema que se está controlando y se generan unas órdenes de mando apropiadas para llevar el sistema a los estados que se deseen.

El concepto de servomecanismo puede abarcar desde dispositivos muy sencillos, hasta sistemas muy complejos en los que se está controlando un proceso muy complicado como puede serlo el de la fabricación de papel en proceso continuo, el control de hornos de producción de cemento, etc.

En cada caso, toda la técnica de control estará contenida en el programa que maneje al sistema, pudiendo ir desde un programa de cálculo sencillo que resuelva una simple ecuación simulando la transmitancia deseada, a todo un complejo sistema de parámetros y tablas que contengan la tabulación de las órdenes que se deben generar para cada situación.

También pueden incluirse aquí los llamados «robots» industriales que son capaces de aprender la operación que ejecuta un operario especializado y repetirla posteriormente con precisión y velocidad mejores que las que conseguiría el mismo operario que enseñó a la máquina.

En cualquier caso, los componentes básicos del sistema son siempre los mismos, pudiendo cambiar enormemente la periferia, transductores, etc. que rodean al microprocesador, siendo las más de las veces mucho más costosos los aparatos de captura de datos que el propio sistema que ejecuta los programas de control.

En el nivel 2 se habla de los sistemas que controlan pequeñas unidades autónomas, como es el caso de las balanzas peso-precio, o las cajas registradoras por ejemplo.

En este tipo de unidades, el microprocesador asume todas las funciones de control. Explora las entradas, teclados, etc., visualiza los valores del peso o del precio, calcula las cantidades a pagar, acumulan las cifras de todo el día, controla los mecanismos de impresión de la factura, apertura de la caja, etc. y asume todos los automatismos que tradicionalmente se realizaban a través de elementos mecánicos o electromecánicos.

En estas unidades el microprocesador suele tener grados de actividad muy bajos, por lo que los requerimientos de velocidad no existen y pueden usarse microprocesadores muy integrados, de baja velocidad y también económicos, ya que les sobra capacidad de ejecución para llevar a cabo funciones tan poco complejas.

Un poco más de complejidad existe si esta unidad autónoma está unida a un sistema central remoto que recoja datos de estas unidades, como puede ser el caso de un conjunto de cajas de un almacén unidas a una unidad básica de recogida de información.

En estos casos el microprocesador controlará también la comunicación con el sistema central. Dicho contro! variará en complejidad según sea una comunicación unidireccional o bidireccional, serie o paralelo, con controles en la información o no, etc., pudiendo soportar el microprocesador el conjunto de operaciones de comunicación además de las operaciones propias de la unidad en cuestión.

El nivel 3 hace referencia a controladores de periféricos. Será en este tema en donde se hará más hincapié porque su importancia es considerable. Un sector muy importante de controladores de periféricos en sistemas informáticos está basado en microprocesadores. Los tipos utilizados son muy diversos ya que en cada caso se pretende la utilización del que mejor se adapta a los requerimientos del controlador, contemplando criterios de velocidad, costo, consumo, utilización de sistemas de desarrollo, etc.

Pero no es solamente una más fácil implantación y menor costo lo que aporta un microprocesador a la hora de diseñar un controlador de periferia, sino que permite al diseñador pensar en un controlador cada vez más inteligente y con mayores características, ya que para ello no necesita complicar la electrónica, sino que tratará de potenciar las rutinas de control escritas para el controlador.

Así pues, la tendencia será mejorar las características, siendo cada vez más autónomos y ocupando menos tiempo de la unidad de proceso a la que estén conectados como periféricos, resultando, de esta forma, potenciado todo el sistema.

6.5.3 Ventajas y desventajas frente a otros sistemas

Los otros sistemas a que hace referencia este apartado serán los diseñados y realizados con circuitos de baja escala de integración o bien los circuitos controladores especializados para cada caso y con escalas de integración elevadas.

Pasemos a enumerar algunas características favorables y desfavorables de estos sistemas. Entre las ventajosas se pueden citar las siguientes.

Versatilidad. Es esta una característica que poseen los controladores basados en microprocesadores, frente a los que están hechos con circuitos integrados sencillos o con circuitos controladores especializados.

Debido a que el comportamiento del controlador ha de ser

el reflejo de su programa de control, es evidente que solamente con cambiar el programa a ejecutar pueden cambiarse las funciones del controlador. Así pues, un diseño siempre puede ampliarse en características reprogramándolo convenientemente, o puede readaptarse, en el caso de que existan modificaciones en el ingenio a controlar.

Por ejemplo, si se trata de hacer un controlador de discos flexibles podrían tomarse dos soluciones, o mejor dicho, tres. La primera consistiría en hacer un diseño con circuitos y puertas convencionales. Esta solución sería de diseño engorroso y haría que el producto resultase caro de fabricación. Por otra parte, sería una solución muy rígida, es decir, cualquier cambio o mejora que se pretendiese introducir supondría un rediseño de circuitos eléctricos, un rediseño de circuitos impresos, etc., proceso de rediseño que resulta muy oneroso.

Una segunda solución sería la de intentar encontrar en el mercado un controlador de disco flexible que viniese integrado en una única pastilla o circuito. Esta manera de resolver el problema resulta ya más practicable, aunque el mecanismo se suele invertir, es decir, una vez encontrado el circuito integrado que más interesa para el diseño del controlador, las características de este último se amoldarán a lo que dicho circuito integrado sea capaz de hacer. En definitiva, el controlador de disco flexible vendrá condicionado en características a lo que ofrezca el circuito específico de control.

Por otro lado, en algunas ocasiones es necesario proveer circuitos lógicos adicionales para adaptar convenientemente el disco flexible en cuestión al circuito elegido como controlador, suponiendo esto, un gasto extra en circuitos integrados.

Esta segunda forma de resolver el controlador adolece de una desventaja análoga a la del primer caso, y es la de su rigidez. Una vez diseñado el controlador, será muy costoso cambiarle o mejorar sus características, ya que la lógica cableada tiene sus costes de cambio elevados.

Por úlimo, la solución de un controlador basado en un microprocesador goza de la ventaja de la que se hablaba al principio, es decir, la versatilidad.

Si el diseñador ha previsto dejar espacio libre en memoria (RAM y ROM) siempre le será posible introducir mejoras en su controlador, o adaptarse a discos flexibles algo diferentes sólo con modificar el programa de control, modificación ésta que no supone costos excesivos en su desarrollo y que no implica apenas gastos en la producción, ya que todo se seguirá fabricando de la misma forma y la única diferencia será que el programa introducido en la memoria fija (ROM) habrá cambiado.

Todo esto hace que el controlador pueda ser mucho más potente que los basados en circuitos muy especializados. Por ejemplo, puede disponer de sistemas muy elaborados de tratamiento de errores, de reintentos, ejecutar funciones especiales, etc., características que pueden ampliarse, no quedando nunca el diseño cerrado a nuevas posibilidades, permitiendo que la vida del producto antes de su obsolescencia sea mayor.

Otro caso podría ser el de intentar conectar el controlador en cuestión a un equipo diferente, con un protocolo distinto, y con diferencias en el manejo de la periferia. Si el controlador se ha diseñado con un microprocesador se tratará de cambiar los programas, y eventualmente alguna modificación electrónica en la interfaz para que pueda conectar el controlador en el nuevo sistema.

Otra posibilidad en cuanto a potencia se refiere, consiste en hacerle funcionar en modo canal. Normalmente, en un sistema informático compuesto de unidad central, memoria y periferia, disponiendo también de accesos directos a memoria, la unidad central debe dar a los periféricos las órdenes oportunas para que éstos sepan cómo y dónde deben hacer sus accesos a memoria. Estas órdenes se suelen dar a través de instrucciones de entrada-salida, y conllevan una cierta ocupación de unidad de proceso, por el simple hecho de tenerlas que ejecutar y enviar a los periféricos. Pero existe un sistema más sencillo y rápido de ejecución. Este sistema será el de que los periféricos funcionen en modo canal.

En este caso, la unidad de proceso no enviará al controlador los datos para su acceso a memoria, sino que estos datos los deposita en la memoria propiamente dicha, enviando al controlador únicamente la primera dirección de memoria en donde encontrará el resto de datos necesarios para concretar el acceso. El controlador deberá acudir a dicha posición, interpretar las órdenes (mandatos) y datos allí depositados por la unidad central. Si estos mandatos y datos le conducen a que se lleve a cabo una transferencia de información de un determinado número de caracteres, el controlador ejecutará dicha transferencia y una vez realizada, examinará si entre los últimos datos recibidos existe alguna orden más que le obligue a seguir haciendo nuevas transferencias, interrumpiendo a la unidad central del sistema, o concluyendo y parando, etc.

De esta forma no sólo son datos lo que la unidad central de proceso envía al controlador a través de la memoria, sino que además le envía órdenes. El controlador deberá, pues, tener capacidad de interpretación y ejecución de dichas órdenes, liberando así al sistema del engorro de que con cada bloque, le tenga que reprogramar vía entrada-salida su siguiente acción.

Pues bien, este funcionamiento del controlador es más sencillo de realizar si se dispone de un microprocesador y se le programa convenientemente para que lo haga.

En resumen, la versatilidad es una de las características más ventajosas de los sistemas de controladores basados en microprocesadores.

Fiabilidad. El hecho de que se use una escala de integración muy alta, como es la de los microprocesadores, reduce el número de fallos frente a sistemas de baja integración y que utilizan mayor número de componentes.

Todo esto repercute en el tiempo medio entre fallos y por tanto en el número de intervenciones necesarias para la reparación del equipo en cuestión.

Bajo consumo. También y por razones de escala de integración, el consumo es menor en estos sistemas de control basados en microprocesadores. Es importante tener en cuenta esta característica, ya que se traduce en otras interesantes características como son el menor tamaño de las fuentes, menores problemas de disipación de calor y por último conmutaciones de intensidad más bajas que reducen los ruidos parásitos inducidos en el sistema.

Bajo costo de producción. También esta característica se desprende de la alta escala de integración. Así el coste bajará por las razones siguientes: menor número de componentes, fuentes más pequeñas, evacuadores de calor más pequeños o innecesarios, menor volumen del conjunto y que será más fácil y económico disponerlo en su lugar físico definitivo.

Hasta aquí se han resumido las ventajas principales. Pero no todo son ventajas. Como contrapartida, diseñar un controlador basado en un microprocesador exige disponer de los elementos de desarrollo adecuados para el tipo de microprocesador elegido.

Estos sistemas son básicamente de dos tipos, unos para el desarrollo de programas y otros para la prueba o simulación del conjunto físico, así como para la grabación y borrado de las memorias de sólo lectura (ROM o PROM) que deban quedar instaladas como contenedoras de los programas definitivos del controlador.

Los programas a introducir pueden escribirse en lenguaje máquina, en lenguaje ensamblador o en un lenguaje de alto nivel.

Si el programa ha de ser poco complejo y en el caso de no disponer de sistemas de desarrollo completo, puede programarse en código máquina, aunque programar en dicho código es muy engorroso y poco recomendable.

Para programas medios, el lenguaje ensamblador es muy interesante, y además permite al diseñador no perder de vista en ningún momento el funcionamiento físico y real del controlador, es decir, es un lenguaje muy próximo a las señales eléctricas, sabiendo el programador hacer corresponder sus instrucciones con las señales y actuaciones del conjunto, al nivel casi de puertas.

El lenguaje ensamblador permite también optimizar los tiempos de ejecución de los programas, ya que pueden usarse instrucciones que usen hábilmente las características del microprocesador.

El uso de este lenguaje hace necesario que el sistema de desarrollo disponga de programa ensamblador para poder pasar del programa fuente al objeto en código máquina.

Este programa ensamblador debe ser suministrado por el fabricante junto con su sistema de desarrollo. Lo que suministra también otras veces el fabricante es un «cross-ensamblador», es decir, un programa que sirve también para ensamblar los programas fuente, pero que se puede ejecutar en máquina de otros modelos existentes en el mercado si el usuario dispone de ellas y las quiere utilizar como sistema de desarrollo.

Figura 6.3 Configuración básica de un sistema de control

Por último, se pueden usar lenguajes de alto nivel, suministrados por el fabricante a la hora de desarrollar el programa de control. Como los tamaños de dichos programas no son excesivamente grandes, puede decirse que estos lenguajes son menos utilizados que el lenguaje ensamblador.

Desde luego, el lenguaje de alto nivel es más potente y por tanto muy útil para programar, aunque el programa objeto que se genera después de la compilación no es tan eficiente en tiempo de ejecución como un programa escrito en ensamblador.

Lo que también posibilita a veces el fabricante es la mezcla de programas fuente escritos en ensamblador con programas escritos en lenguaje de alto nivel, para que así se puedan utilizar ambas ventajas, es decir, la de programación eficiente con el ensamblador y la de programación sencilla con el lenguaje de nivel elevado.

Todo esto conduce a la necesidad de disponer de los sistemas de desarrollo «hardware» y «software» apropiados y que deben compararse normalmente al mismo fabricante que suministra el microprocesador utilizado. Esto significa desventaja económica en el desarrollo del controlador, aunque es fácil su amortización, ya que dichos sistemas de desarrollo tampoco son excesivamente caros, como para resultar prohibitivos.

6.5.4 Configuración básica

En la figura 6.3 puede verse una configuración básica de un sistema de control basado en un microprocesador. Una línea de puntos separa los elementos más usuales o básicos de los de menor utilización.

Entre los básicos citaremos los siguientes:

6.5.4.1 *Microprocesador*

Es el elemento básico del sistema y el que marca la pauta sobre el funcionamiento del bus, su protocolo y las señales específicas de su uso. La arquitectura del sistema montado dependerá del microprocesador. Por ejemplo, si admite interrupciones, si puede llevar más o menos memoria, como se organiza ésta, qué longitud de palabra tiene todo el conjunto, etc., aunque en la realidad la arquitectura de los diferentes sistemas es muy parecida a la del modelo de la figura.

6.5.4.2 *Memoria del sistema*

La memoria está constituida por una parte de memoria de acceso aleatorio (RAM) y otra parte de memoria de sólo lectura (ROM). El tamaño de la memoria dependerá del programa que deba contener para realizar la aplicación.

Esta es una de las diferencias principales entre un sistema informático realizado con un microprocesador y un sistema de control basado en un microprocesador también. En los sistemas informáticos prácticamente la totalidad de la memoria instalada es memoria de acceso aleatorio y una parte muy pequeña de memoria se instala como memoria de sólo lectura (ROM), ya que el programa de arranque apenas debe estar contenido en memoria fija (ROM) y los programas siguientes del sistema —programas básicos y/o de aplicación— serán cargados en la memoria de acceso aleatorio (RAM) desde memorias externas, tales como

discos flexibles, casetes, discos, etc., o algún otro tipo de soporte externo de información.

El programa contenido en memoria fija se ejecutará sólo al principio y luego los restantes programas serán ejecutados desde la memoria de acceso aleatorio. En esta memoria estarán también las áreas de datos necesarios para dichos programas.

En cambio, en un controlador basado en un microprocesador, todos los programas residen normalmente en memoria fija ya que ellos mismos van a ser fijos. En concreto, sólo existirán en memoria los programas específicos de control y ninguno más. Además, de esta manera, el conjunto resulta autónomo y no necesita elementos externos que le carguen el programa de trabajo. Así pues, la memoria aleatoria resulta innecesaria a efectos de contener programas. No obstante, se debe disponer de una cierta cantidad de dicha memoria para que los programas la utilicen como zona de trabajo de sus datos. A veces, en sistemas poco complejos, los acumuladores internos del procesador pueden ser suficientes para este fin prescindiendo totalmente de la memoria aleatoria.

A pesar de todo, mientras se están escribiendo y desarrollando los programas de control es conveniente disponer de memoria aleatoria y ejecutar los programas desde esa memoria. Cuando los programas quedan depurados y definitivos se les graba en memoria fija. En ese momento, en el que el diseño de programas queda concluido, ya no es necesaria la memoria aleatoria a efectos de ser usada como zona de programa.

6.5.4.3 *Vías de acceso de entrada-salida*

A través de estas vías de acceso o puertos (en inglés, ports) o registros el microprocesador recibirá información de las partes del controlador y por donde emitirá las órdenes de mando a los diferentes actuadores del mismo.

En general cualquier elemento que maneje información del sistema deberá estar conectado a uno de estos puntos con las interfaces pertinentes en cada caso.

Aquí entrarán las medidas analógicas, de tensiones, intensidades, temperaturas, posición, etc., que deberán ser previamente convertidas a señales digitales. Por estos puntos saldrán órdenes de manejo de motores, válvulas, relés, etc., y en general cualquier dato externo a la unidad central.

6.5.5 Configuración ampliada

Además de los elementos básicos, pueden aparecer otros que amplíen la configuración básica antes indicada, ya no como periféricos, sino como elementos de apoyo al microprocesador.

6.5.5.1 *Temporizador de intervalos*

Se maneja con órdenes de entrada-salida y se puede considerar como un periférico. No obstante, a veces viene integrado con el microprocesador ya que su utilización es muy frecuente. No es un elemento imprescindible porque siempre pueden efectuarse temporizados por programas escribiendo bucles más o menos largos, pero como temporizar de esta manera ocupa al microprocesador, se prefiere en muchos casos disponer de temporizador de intervalos automático que avise con una interrupción después de temporizar. Esto exige, por otra parte, que el microprocesador disponga de los mecanismos físicos necesarios para poder soportar interrupciones.

6.5.5.2 *Sistemas de interrupciones*

Tampoco son esenciales estos elementos. Su función puede llevarse a cabo por programas de consulta que analicen si ciertos eventos han tenido lugar, en vez de esperar a que dichos eventos causen interrupción o aviso.

Claro está que la exploración por parte de la unidad de proceso roba tiempo de ejecución a los programas en curso, restando potencia efectiva al conjunto.

Si el o los programas de control no ocupan demasiado al microprocesador, puede encargarse éste de los trabajos de temporizar y auscultar el estado del sistema. Si, por el contrario, los programas de control son complejos y largos, conviene apoyar con más elementos al microprocesador descargándole de las tareas citadas.

Los microprocesadores del mercado no suelen disponer de muchas vías de entrada de interrupción y en este caso es conveniente poner circuitería accesoria para aumentar el número de dichas vías. Los sistemas de interrupciones cumplen este cometido.

6.5.5.3 *Otras conexiones*

Se mencionan aquí tipos menos frecuentes de conexiones al sistema. Así, por ejemplo, no es usual que en sistemas de control se instalen circuitos de acceso directo a la memoria conectada al bus aunque el microprocesador tenga capacidad para soportar el dar control de sus buses a otros circuitos. Tampoco son usuales comunicaciones con otros buses de otros microprocesadores, formando sistemas de multiprocesador. Menos raros, no obstante, son los circuitos que saben comunicarse en serie síncrona o asincronamente con otros procesadores, circuitos que aunque se manejan con instrucciones de entrada-salida suelen ir conectados directamente al bus del microprocesador, ya que están hechos con ese fin.

En resumen, lo más usual será ver la configuración descrita como básica, ampliada como mucho con temporizador o interrupciones, aunque eso no obsta para que puedan diseñarse sistemas muy complejos basados en μP.

6.5.6 Criterios de selección

Varios son los criterios que deben presidir la elección de un microprocesador de entre la extensa gama que el mercado ofrece, aunque todos ellos pueden clasificarse en dos grupos principales. Los criterios de orden técnico y los de orden económico.

El principal criterio de orden técnico es la velocidad. El microprocesador debe tener suficiente velocidad de ejecución para trabajar en tiempo real con los mecanismos que gobierna y controla. Es decir, los resultados de los programas deben actuar a tiempo sobre los citados mecanismos. Veamos un ejemplo. Supongamos que debemos poner un microprocesador para controlar un disco flexible. Uno de los programas del controlador será el que lleve a cabo la función de lectura de un determinado número de sectores. En cada sector deberá leer las cabeceras correspondientes, trasportar la información contenida en el sector a la zona de memoria correspondiente —se le supone con acceso directo a memoria— y por último, deberá hacer los chequeos pertinentes de los errores que puedan haberse producido en la lectura de los datos.

Dentro de la misma pista habrá un número determinado de sectores para los que el citado programa de lectura efectuará las mismas operaciones que con el primer sector.

Pues bien, conviene que el programa haya sido lo suficientemente veloz para ejecutar todo lo correspondiente a un sector antes de que llegue información del sector siguente, pues de lo contrario, con cada lectura de sector el disco flexible perdería vuelta ya que si un nuevo sector empezase a ser leído por los cabezales de lectura y el programa de control no hubiese concluído con el manejo del sector precedente, sería necesario esperar a que el disco flexible completase la vuelta y volviese a poner el sector en cuestión bajo los cabezales de lectura encontrando listo al microprocesador para atenderlo.

Igual sucede a nivel de carácter leído. Si el programa de recoger un carácter y depositarlo en memoria fuese tan lento de ejecución que no llegase a tiempo de recoger el carácter siguiente, resultaría que el sistema no serviría por cuestiones de velocidad.

Dos son las variables que intervienen en la velocidad de ejecución de un programa. Una de ellas es la velocidad intrínseca del micrôprocesador. Es decir, la velocidad con que se ejecuta cada una de las instrucciones del programa. La otra variable es el número de instrucciones a ejecutar que lleva el programa en cuestión. Nos referimos aquí al número de pasos a ejecutar, no al número de instrucciones programadas, ya que las instrucciones que sirven para programar bucles generan muchos pasos de programa cuando se ejecutan.

Estas dos variables se pueden hacer óptimas. Evidentemente la primera se mejorará eligiendo un sistema más veloz, es decir, de menor tiempo de ciclo. Esto se logrará con un procesador de tiempo de ejecución menor y con unas memorias de bajo tiempo de acceso, así como utilizando circuitos de interrupción, temporizador, etc., que permitan descargar al microprocesador.

La segunda variable se mejorará optimizando los programas y evitando la ejecución de operaciones redundantes. La optimización debe empezar por el análisis del programa de control, optimizando el organigrama. La programación puede ser en lenguaje de alto nivel o en ensamblador. En el primer caso no se tiene control sobre el programa resultante, ya que el compilador generará un programa objeto que probablemente contendrá más instrucciones máquina que las imprescindibles, por lo cual, cuando se quieren mejorar los tiempos de ejecución, es recomendable usar lenguaje ensamblador en los programas.

Aparte del problema de la velocidad, el microprocesador deberá ser capaz de soportar la cantidad de memoria requerida, así como de que sea suficiente el número de «puertos» que puede tener conectados.

Entre los criterios de orden económico se pueden citar los siguientes. En primer lugar el coste del propio sistema de microprocesador memoria y periferia. Hay que pensar que si el número de equipos a producir es grande, será conveniente que el coste unitario sea bajo ya que en eso estribará una parte del coste del controlador.

En este punto conviene considerar los sistemas altamente integrados que incorporán en una única pastilla o circuito el microprocesador, una determinada cantidad de memoria EPROM (memoria de sólo lectura, borrable y reprogramable) y una determinada cantidad de memoria de acceso aleatorio. (RAM), así como varias líneas de entrada-salida, sistemas de interrupción, temporizador de intervalos, etc.

Compatible con bus de	Suministrador	Modelo	Velocidad	ESPECIFICACIONES ROM	RAM	N.º bits puertos	N.º bits temporiz.	Tecnología	Precio disponibilidad	Comentarios
8080	Intel	8041 8741 UPI	1-6 MHz	1 K×8	64×8	8,8,2	8	N-MOS 5 V 40 patas	56 dól. YA	Versión «esclavo» del 8048
8080	National	420	4 μs ciclo instr.	1 K×8	64×8	23	—	N-MOS 5 V 28 patas	2 dól. YA	Casará directamente con el microbús 8080 de National, y servirá como periférico
F8	Mostek Fairchild	3870	4 MHz	2 K×8	64×8	8,8,8,8,	8	N-MOS 5 V 40 patas	10 dól. YA	Cualquier único chip como el 3870 puede servir como control.
9900	TI	9940	5 MHz	2 K×8	128×8	32 líneas serie	14	N-MOS 5 V 40 patas	2 dól. YA	Dos versiones: -EROM -ROM
CP1600 General	GI	1645	1 MHz 4 μs instr.	256×12	28×8	4,4,4,	Temporizador/contador	N-MOS 5 V 18 patas	2 dól. YA	Tiene interrupción para entradas externas y temporizador
General	Rockwel	PPS-4/1	80 kHz	640×8 1344×8 2048×8	48×4 96×4 128×4	4, 4, 4, 4 2 serie	Temporizador	P-MOS 42 patas	5 dól. YA	Disponibles muchos modelos preprogramados
General	National	SC/MP	2-4 MHz	—	4×8 registros de la CPU	8,2,3 y flags	Instrucción especial de temporizar	PMOS N-MOS 40 patas	9 dól. YA	Con controles para configuraciones maestro-esclavo
General	Signetics	8×300	250 ns ciclo de instr.	Extern.	7×8	Maneja E/S	—	Schottky 50 patas	32 dól. YA	Muy usado como controlador rápido

Tabla 6.4 Controladores basados en microprocesadores

Estos circuitos son más caros, pero en caso de servir para los propósitos del diseñador ahorran costes de producción tanto en circuitos integrados como en circuitos impresos, mano de obra, etc.

Otro factor de coste que debe considerarse es el del sistema de desarrollo de programas. Antes de grabar los programas en memoria fija (ROM) deben ejercitarse, probarse, etc., en memoria aleatoria (RAM), deben ser editados y se debe disponer de los sistemas necesarios para introducirlos en la memoria. El sistema de desarrollo puede ir desde una plaquita con un teclado, para programas pequeños o todo un sistema de procesador, memoria, teclados, pantalla, disco flexible, impresora, con sus sistemas operativos, compiladores, etc., para programas más complejos.

Todos estos sistemas de desarrollo, orientados a una gama de microprocesadores de una firma determinada, tienen un precio que en muchos casos determina el tipo de procesador que va a ser utilizado.

Por ejemplo, si ya se dispone de un sistema de este tipo procedente de diseños anteriores, es muy probable que se elija un microprocesador compatible con el sistema de desarrollo.

Por último, cabe considerar otro factor muy influyente como es el elemento humano. Cuando el técnico está familiarizado con un determinado microprocesador, es muy probable que para nuevos diseños elija el mismo o alguno similar, siempre y cuando este microprocesador cubra los mínimos requerimientos de características necesarias para el controlador.

En la tabla 6.4 se muestran los circuitos integrados que incluyen además de un microprocesador, otras partes como son, memorias, interrupciones, temporizadores, etc.

6.6 APENDICE: CONTROLADOR DE INTERRUPCIONES 8259

6.6.1 Descripción funcional

El 8259 es un circuito diseñado especialmente para manejar interrupciones en tiempo real en un sistema basado en microprocesador. Puede gestionar hasta ocho niveles de interrupción y está diseñado de tal manera que permite su interconexión con otros 8259 hasta formar un conjunto de ocho circuitos con capacidad de manejo de 64 interrupciones de niveles diferentes.

Todas sus formas de trabajo pueden programarse vía entrada-salida como si se tratase de un periférico. Puede efectuarse una programación de diferentes algoritmos de prioridad para las diferentes vías de interrupción, pudiendo además cambiarse dinámicamente dichos algoritmos de prioridad durante la ejecución del programa principal, adecuándose en cada momento a las necesidades del sistema.

En la figura 6.4 se representa un diagrama de bloques del 8259. Pasemos a describir cada uno de los bloques de la figura.

6.6.2 Registro de interrupciones (IRR) y registro de la interrupción en servicio (ISR)

Las vías de llegada de interrupciones procedentes de los periféricos (IR0-IR7) se tratan a través de dos registros en cascada, el registro de interrupciones (IRR) y el registro de la interrupción en servicio (ISR). En el primero (IRR) se almacenan todos los niveles de interrupción que están

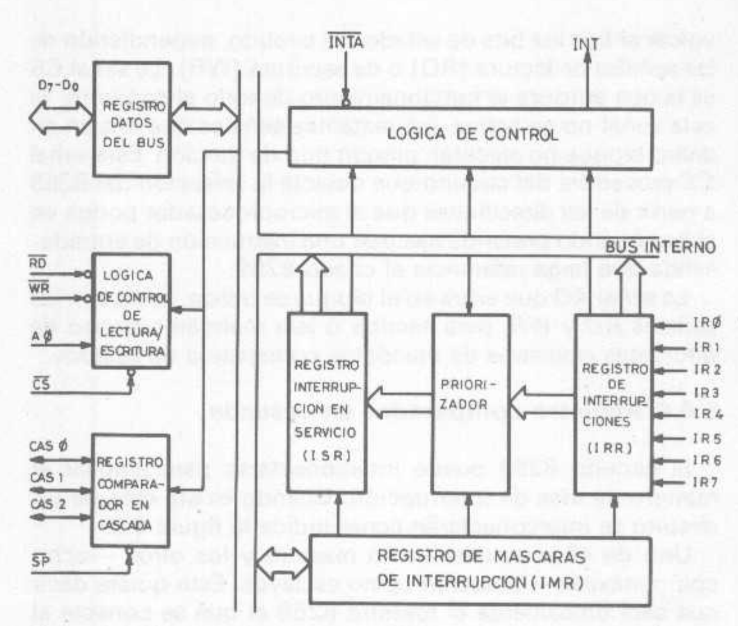


Figura 6.4 Diagrama de bloques del 8259. IR (0 ÷ 7) = Peticiones de interrupción (de 0 ÷ 7); INT = Interrupción (hacia el microprocesador); INTA = Interrupción reconocida (por el microprocesador); D7 ÷ DO = Datos del bus del procesador; RD = Señal de lectura de datos; WR = Señal de escritura de datos; CS = Señal de selección del circuito; cas0 – cas3 = Señales usadas para ampliar en cascada el sistema; SP = Señal que indica si el circuito es maestro o es esclavo; AO = Ver descripción.

solicitando servicio y en el segundo se registra el nivel de interrupción que está siendo atendido.

Con el flanco positivo de alguna interrupción entrante se activa la línea de salida INT y la interrupción debe mantenerse hasta que sea aceptada por el microprocesador con la señal INTA. Esta señal desactiva al registro de interrupciones y activa al registro de interrupción en servicio que sólo deberá ser desactivado al final de la atención de la interrupción, bien sea de una forma automática o bien a través de una instrucción de entrada-salida enviada por el manejador de dicha interrupción.

6.6.3 Priorizador

Este bloque lógico sirve para determinar cuál de las posibles entradas de interrupción activas debe ser atendida en ese momento. La interrupción que este bloqueo determine será la que pase al «registro de interrupción en servicio».

Como ya se ha dicho, una vez resuelta la prioridad entre varias posibles interrupciones se avisa —con la señal INT— al procesador y este reconoce la interrupción contestando con la señal INTA. El microprocesador genera más de una señal INTA con el fin de que el 8259 libere hacia el bus de direcciones la dirección de inicio de la subrutina a donde bifurcarse.

6.6.4 Registro de datos del bus

Es un registro bidireccional tres-estados que hace de interfaz con el bus del microprocesador. Todas las palabras o mandatos de control para el 8259, así como los bits de estado, se transfieren a través de dicho registro.

6.6.5 Lógica de control de lectura-escritura

La función de este bloque es aceptar mandatos para programar el modo de funcionamiento, así como hacer

volcar al bus los bits de estado del circuito, dependiendo de las señales de lectura (RD) o de escritura (WR). La señal CS es la que autoriza el funcionamiento de todo el conjunto. Si esta señal no se activa, los restantes señales que entran en dicho bloque no ejecutan ningún tipo de función. Esta señal *CS* provendrá del circuito que detecta la selección del 8259 a partir de las direcciones que el microprocesador ponga en el bus cuando pretenda ejecutar una instrucción de entrada-salida que haga referencia al citado 8259.

La señal *AO* que entra en el bloque se utiliza, junto con las señales *RD* y *WR*, para escribir o leer mandatos dentro de diferentes «registros de mandato» o «registros de estado».

6.6.6 Registro comparador en cascada

El circuito 8259 puede interconectarse para ampliar el número de vías de interrupción. Cuando exista más de un circuito se interconectarán como indica la figura 6.5.

Uno de ellos actuará como maestro y los otros —ocho como máximo— actuarán como esclavos. Esto quiere decir que será únicamente el maestro 8259 el que se conecte al microprocesador para interrumpirle. La salida de interrupción (INT) de todos los esclavos 8259 se conectará a las vías de entrada de interrupción del maestro, haciendo trascender la interrupción al microprocesador a través de él.

Para hacer posible este sistema de interconexión, el circuito dispone de dos medios, uno la señal *SP*, y otro el registro comparador en cascada.

Con la señal *SP* se programa que el 8259 actúe como maestro o como esclavo según se conecte a «uno» o a «cero».

El registro comparador en cascada tiene tres conexiones (CAS0-CAS2) con el exterior por las que el 8259 impone un código si actúa como maestro o recibe un código si lo hace como esclavo.

En ambos casos el código corresponde al del 8259 que va a ser tratado por el microprocesador, es decir, aquél que ha recibido la interrupción más prioritaria —no enmascarada—.

6.6.7 Funcionamiento

Las características que más potencian al 8259 son la posibilidad de ser programable su forma de trabajo, y la utilización de la instrucción de bifurcar a la rutina de atención aportando la dirección correspondiente.

La secuencia normal de acontecimientos en una interacción 8259-microprocesador es la que sigue:

1. Alguna o varias líneas de interrupción se activan indicando al 8259 que algún periférico está solicitando atención del microprocesador.
2. El 8259 acepta la interrupción, la prioriza y envía una señal de interrupción al microprocesador.
3. El microprocesador reconoce la interrupción devolviendo una señal de reconocimiento (INTA).
4. Con la recepción de dicho impulso (INTA), el 8259 imparte un código de instrucción de bifurcar en el bus de datos.
5. La instrucción generada hace que el microprocesador genere dos señales (INTA) adicionales hacia el 8259.
6. Estas dos señales hacen que el 8259 suministre al bus dos octetos adicionales que serán la dirección de bifurcación a la rutina de atención. El contenido de dichos octetos dependerá de lo que originariamente se haya programado en el 8259, vía entrada-salida.
7. Con esto se completan los tres octetos que requiere la instrucción de bifurcar a subrutina. El registro de interrupción en servicio no se borra hasta que, en el final de la subrutina, se envíe un mandato de final de interrupción.

La secuencia descrita ocurre cuando el microprocesador es el 8080. En el caso de tratarse del 8086, el 8259 sólo debe suministrar un octeto de dirección pudiendo además apagarse el registro de interupción en servicio de una forma automática sin que sea necesario programar un borrado de este registro en el final de la rutina de atención.

6.6.8 Programación

El 8259 acepta dos tipos de mandatos procedentes del

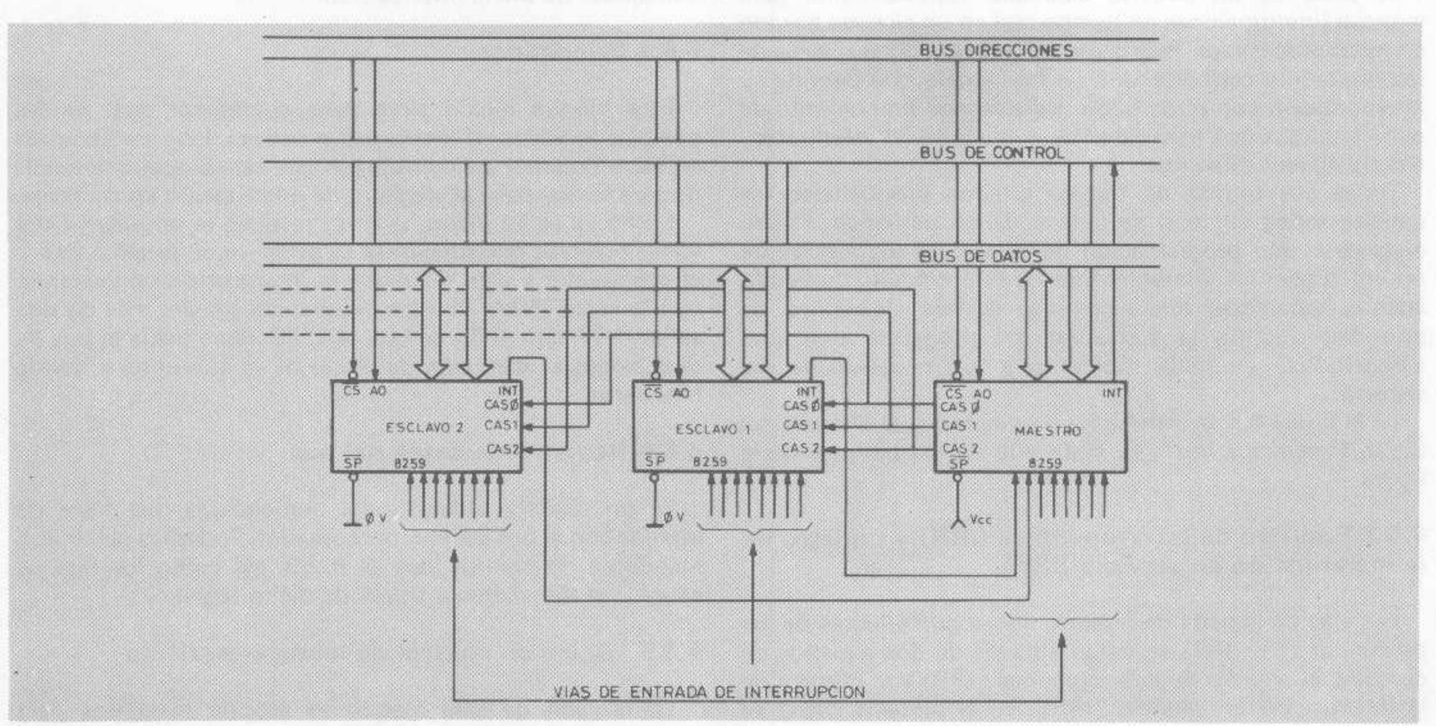


Figura 6.5 Registro comparador en cascada

microprocesador; 1) Mandatos de inicialización y 2) mandatos de operación.

Con los primeros se prepara al controlador en el punto inicial en el que queda dispuesto para aceptar interrupciones.

Con los segundos se programa al 8259 para que pueda trabajar en distintos modos de interrupción. Los modos son:

— Modo de prioridad según posición física o patilla de entrada.

— Modo de prioridades rotativas

— Modo de máscara especial

— Modo de barrido.

Vamos a describirlos a continuación viendo antes las máscaras de que dispone el 8259.

6.6.9 Máscaras de interrupción

Cada vía de interrupción puede ser enmascarada individualmente programando de forma adecuada el registro de máscaras de interrupción (IMR). Esta programación se hará con un mandato de operación.

El IMR opera tanto sobre el registro de interrupciones como sobre el registro de interrupción en servicio. Cuando una interrupción está siendo atendida —está registrada en el ISR— quedarán inhibidas las interrupciones de nivel inferior. Si se quiere no obstante atenderlas, pueden hacerse dos cosas. Una de ellas consiste en enviar al 8259 una instrucción de fin de interrupción para que borre el registro ISR. La otra es usar el modo máscara especial programándolo con un mandato de operación y funcionando como se verá después.

6.6.10 Modo de prioridad según posición física

En este modo opera el 8259 después de habérsele inicializado con un mandato de inicialización. En este modo las entradas de interrupción quedan ordenadas en prioridades de 0 a 7. Cuando ocurra una interrupción del microprocesador, se enviará la dirección de subrutina desde el 8259 de forma que se atienda la más prioritaria de entre todas las interrupciones presentes. Se activa el registro de interrupción en servicio y sólo se apaga por una instrucción de fin de operación. Mientras esté activo ninguna interrupción menos prioritaria se dejará pasar. No así con las de nivel superior que sí estarán permitidas.

La entrada o patilla más prioritaria es IRO y la de menor prioridad IR7.

6.6.11 Modo de prioridades rotativas

Se utiliza cuando se quiere asignar una prioridad igual a todas las entradas, por ejemplo las procedentes de canales de comunicación. El 8259 proporciona dos submodalidades:

a) Prioridad rotativa automática

b) Prioridad rotativa específica

En el primer submondo, después de haber sido atendido un periférico pasa a tener la prioridad más baja, de forma que en el peor de los casos, si solicita una nueva interrupción deberá esperar a que las otras siete se atiendan. Esto en el supuesto de que todas las otras siete hayan solicitado atención del microprocesador.

En el segundo submodo, el programador puede cambiar las prioridades asignando a una de las entradas la última prioridad. Así por ejemplo, asignando a la entrada número 5 el último nivel de prioridad, la entrada número 6 quedará sin más, como la de prioridad más alta.

6.6.12 Modo máscara especial

Este modo se usa cuando algunas interrupciones están enmascaradas por el registro de máscaras IMR. Si por alguna razón están ejecutándose alguna subrutina que está enmascarada (puede ocurrir cuando la propia subrutina quita intencionadamente su máscara) se puede permitir a los niveles de prioridad inferior el que interrumpan si se está funcionando en modo «máscara especial». Los niveles de prioridad más alta no son afectados por funcionar en dicha modalidad.

6.6.13 Modo barrido

En este modo el procesador no permite ninguna interrupción. El servicio de atención a los periféricos se realiza porque el programador lo solicita a través de un mandato «Poll».

Cuando el 8259 recibe una instrucción de salida —escritura— acompañada de un mandato «Poll», se prepara para que en el transcurso de una instrucción de entrada —lectura— que debe ser enviada a continuación se considere como si se tratase de un reconocimiento de interrupción y, si en ese momento hubiese una petición de interrupción en alguna entrada, se volcaría un octeto al bus de datos que tendría, en su bit de menos peso, un «uno» en el caso de haber una interrupción y un «cero» si no lo hubiese. En el caso de haber más de una, aparecería en dicho octeto el código de la más prioritaria.

Este modo se usa en los casos en que las rutinas de atención son iguales para la mayoría de los niveles. De esta forma se ahorra espacio en memoria fija (ROM) al no tener que repetir la rutina. Otra aplicación de este modo es aquella en la que se quiere ampliar a más de 64 niveles el número de interrupciones del microprocesador.

6.7 APENDICE: CONTROLADOR DE ACCESO DIRECTO A MEMORIA DMAC-M6844

6.7.1 Introducción

El controlador de acceso directo a memoria DMAC-6844 efectúa la función de transferir datos directamente entre memoria y los controladores periféricos. Controla los buses de datos y de direcciones, igual que el microprocesador, en sistemas basados en la familia 6800 de Motorola.

La interfaz del 6844 con el bus incluye toda la lógica de selección, interrupción, lectura, escritura, etc. para permitir la transferencia de octetos por el bus de datos. La estructura interna contiene los circuitos necesarios para controlar y manejar cuatro canales independientemente configurados. Tiene también registros de control programables, independientes en donde se puede escribir la dirección de inicio de transferencia, longitud de transmisión, así como el modo de configuración y control, prioridad de servicio, control de interrupciones y encadenamiento de datos. Los bits de estado y las líneas de control permiten manejar a los controladores periféricos.

El modo de transferencia de cada canal puede ser programado como de «robo de ciclo» o en modo «bloque».

El ritmo de transferencia que puede soportar es de hasta 1 Mbyte por segundo.

El esquema de bloques del 6844 puede verse en la figura 6.6. En la figura 6.7 puede verse cómo se inserta en un

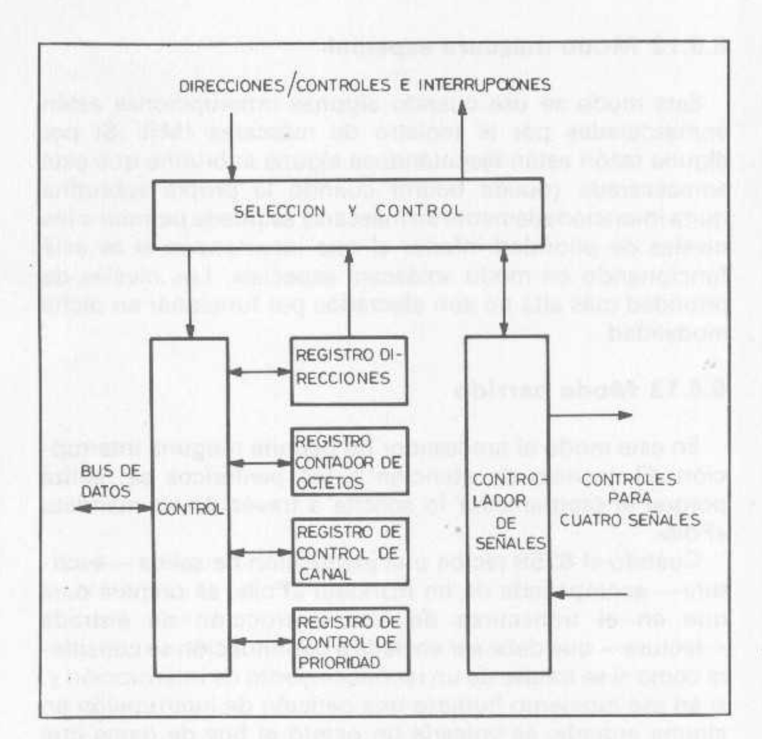


Figura 6.6 Controlador de DMA M6844 (DMAC)

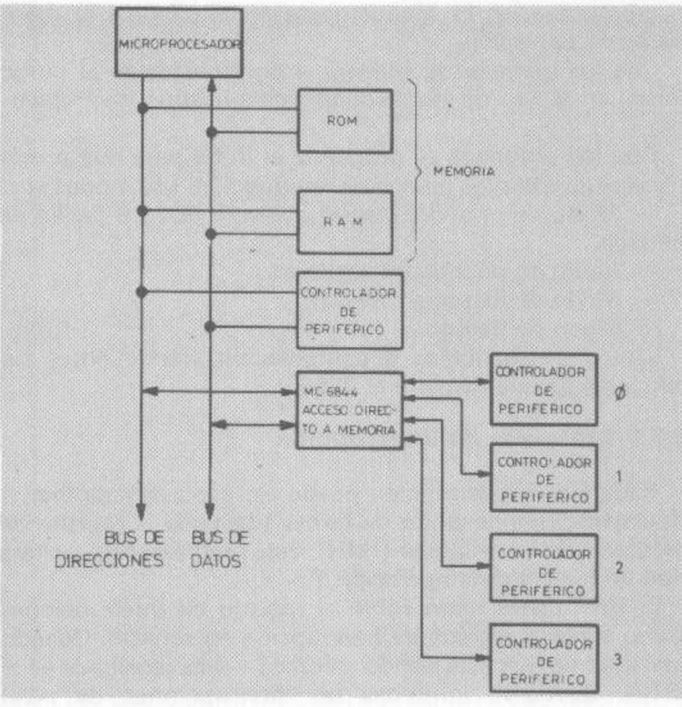


Figura 6.7 Esquema de bloques del sistema 6800

Figura 6.8 Controlador de DMA M6844 (DMAC)

sistema basado en el microprocesador 6800 de Motorola. Por último en la figura 6.8 se representa un esquema de bloques con mayor detalle del citado circuito.

6.7.2 Operación

El circuito que nos ocupa tiene quince registros direccionables, ocho de los cuales son registros de 16 bits. Cada canal tiene su propio registro de dirección y su registro contador de octetos, siendo ambos de 16 bits de longitud. También hay cuatro registros de control por cada canal. Por otra parte existen tres registros más que se usan para control y que son comunes a los cuatro canales. Estos son: el registro de control de prioridad, registro de encadenamiento de datos y registro de control de interrupciones.

Cuando se quiere preparar un acceso a memoria en uno de los canales, el registro de direcciones se deberá cargar con la dirección inicial de memoria —dirección de octeto—, y el registro contador, con el número de octetos que se quieran transmitir.

Los bits en el registro de control de canal establecen la dirección del que transfiere, el modo de transferencia y si en contador de octetos transmitidos debe incrementarse o decrementarse después de cada ciclo. Cada canal puede prepararse para que transfiera en una de las tres modalidades siguientes:

a) Robo de ciclo con bus en «tres-estados»
b) Parada y robo de ciclo
c) Parada y transmisión en modo bloque

Luego se verá con más detalle cada una de estas modalidades.

En el registro de control de canal aparecen dos bits más que indican el estado de éste. Uno de ellos dice si el canal está haciendo algún acceso a memoria y el otro indica si la transmisión ha sido completada.

El registro de control de prioridad recibe las peticiones de transferencia provenientes de los controladores de periferia y establece entre ellos una prioridad de atención que puede ser fija —según la vía física de entrada de cada controlador de periférico— o rotativa —el último atendido pasa a ser el de menos prioridad—.

Cuando se ha completado un canal en una transferencia —su contador ha quedado a cero—, se avisa al controlador de periférico con una señal de «fin de transferencia» y se da una interrupción al procesador.

El encandenamiento de transferencias es controlado por el registro de encadenamiento de datos. Si está en funcionamiento, el contenido del registro de direcciones y del registro contador del canal «3» se coloca dentro de los correspondientes registros del canal seleccionado para que se encadene la siguiente transferencia, cuando el citado canal concluya con la transferencia anterior. Esto permite leer o escribir bloques de memoria a través de un canal de forma repetitiva y sin demora entre bloque y bloque.

El controlador DMAC M6844 suministra las señales de control de memoria durante el ciclo. Cuando un controlador de periférico desea una transferencia directa a memoria, la solicita al circuito controlador, el cual resuelve las prioridades correspondientes y solicita una demanda de acceso directo. Una vez que el controlador recibe el correspondiente reconocimiento, lo pasa al controlador del periférico que lo ha solicitado al mismo tiempo que es transferido el dato. Cuando el contador de octetos transferidos llega a cero, el DMAC avisa al controlador del periférico del fin de la transmisión y a su vez genera una interrupción para el microprocesador.

En el borrado de puesta en marcha, el DMAC se inicializa poniendo a cero todos los registros excepto el de direcciones y contadores. Esto inhibe todas las posibles demandas de transferencias y de encadenamiento de ciclos. También enmascara las interrupciones. Así, antes de que se deje trascender ninguna solicitud de acceso a memoria, deberán ser programados los registros de direcciones y de control y luego permitir el paso de solicitudes de transferencia en el registro de control de prioridad.

6.7.3 Modos de transferencia

Son tres las maneras posibles de llevar a cabo el intercambio de información con el circuito aquí descrito. No obstante, dos de ellas, robo de ciclo con bus en tres-estados y parada y robo de ciclo, se efectúan robando ciclos de memoria al procesador. En ambos casos se solicita un ciclo al microprocesador y éste lo concede poniendo los buses en «tres-estados» y dejando que el DMAC suministre las señales necesarias para la memoria. En el primer caso el microprocesador no se detiene durante el ciclo de acceso y en el segundo caso sí, aunque reanuda su funcionamiento una vez se termina el ciclo.

El tercer método «parada y transmisión en modo bloque» es similar a la de «parada y robo de ciclo» excepto que el número de octetos transmitidos es más de uno. El microprocesador se detiene mientras se transmite todo un bloque de datos. Cuando el contador de octetos llega a cero, la transferencia queda hecha completamente y se devuelve el control al microprocesador. Este modo es el que proporciona mayor ritmo de transferencia de datos, a expensas claro está, de que el microprocesador permanezca inactivo durante todo el tiempo que dura la transferencia.

Los quince registros del DMAC que aquí se describe, pueden ser leídos y/o escritos aunque algunos de los bits de estado están sólo para ser leídos.

6.7.4 Registros de direcciones

Cada canal tiene su propio registro de direcciones de 16 bits. Antes de lanzar una transferencia directa a memoria, se debe cargar en este registro la dirección inicial del bloque a transmitir. Esta dirección se incrementará o decrementará, en los sucesivos ciclos de acceso a memoria, según se haya especificado en el registro de control de canal (bit 3).

6.7.5 Registros contadores de octetos

Cada canal tiene también su propio registro contador de octetos. También este registro deberá ser cargado antes de empezar una transmisión con el número de octetos que se quieren transmitir. La capacidad máxima de transferencia

Bit 1	*Bit 2*	*Modo de transferencia*
0	0	Microprocesador parado y robo de ciclo por carácter
0	1	Microprocesador no parado y robo de ciclo por carácter (buses en «tres-estados»)
1	0	Microprocesador parado y transferencia de un bloque de carácteres
1	1	Ilegal (no debe programarse)

Tabla 6.5 Modos de transferencia y su programación. Circuito M6844 (DMAC)

será pues de 65.536 octetos de datos. Este registro contador es decrementado al principio de cada ciclo de acceso a memoria.

6.7.6 Registros de control de canal

El control de las transferencias de cada canal debe ser programado dentro del «registro de control de canal».

Programación de lectura/escritura. Un bit (0) controla la dirección de transferencia. Si este bit vale uno, el controlador lee en la memoria. Si el bit vale cero se efectúa una escritura en memoria.

Programación de modo bloque/robo de ciclo. Programando uno de los bits (1) del registro se selecciona uno de los dos modos de transferencia. El bit alto implica que se ha seleccionado transferencia en modo bloque.

Programación del funcionamiento en robo de ciclo. Dando un valor 1 al bit 2 se efectúa una selección del modo «tres-estado» (sin parar al procesador) y dando un valor cero a dicho bit la transferencia en robo de ciclo se efectuará en modo: «parada y robo de ciclo».

Los dos bits anteriores sirven pues, para seleccionar uno de los tres posibles modos de funcionameinto. En la tabla 6.5 se muestra su actuación.

Control de incrementado o decrementado de la dirección de memoria. Con la programación del bit 3 se controla el cambio de dirección de transferencia para cada ciclo. Si este bit 3 vale cero, la dirección de transferencia se incrementará cada vez que se ejecute un ciclo —y se descuente el contador de octetos—. Al contrario, ocurrirá un decremento en la dirección de memoria con cada ciclo, si dicho bit 3 se ha programado con valor uno.

Indicador de ocupado/listo. Este bit (6) es un «estatus» que da infrmación de si está ocurriendo una transferencia en el canal consultado, y solo puede leerse, no programarse. Si vale uno, indica «ocupado» y si vale cero, indica «listo» para poder programar una transferencia. Si realmente se programa un acceso a memoria, este bit se pone a uno automáticamente y no cae a cero hasta que la transferencia concluya y el circuito genere la interrupción del procesador y el aviso de fin de transferencia al controlador de periférico.

Indicador de transferencia concluida. Bit 7. Este bit indica que la transferencia de todo un bloque ha sido concluida. Al mismo tiempo que se desactiva el bit de ocupado (bit 6) se activa el bit 7. Cuando el microprocesador lee el registro de control de canal, se apaga automáticamente este bit. También este bit causa interrupción en el microprocesador, si en el registro de control de interrupciones se ha programado así.

6.7.7 Registro de control de prioridades

En este registro se autorizan o inhiben las demandas de transferencia así como la priorización de éstas. Veamos ahora las distintas funciones a través del análisis de los bits de este registro.

Bits de autorización de solicitud de transferencia. (Bits 0 al 3). Cada canal puede ser autorizado a solicitar transferencias con memoria, independientemente de si lo están o no los demás canales. Si el bit correspondiente se ha programado a uno, las demandas de transferencia del canal serán atendidas y si vale cero, serán prohibidas.

Bit de selección del modo de prioridad. (Bit 7). La rutina de servicio de prioridad en el M6844 puede ser seleccionada con este bit. Si su valor es cero, las prioridades quedan en modo fijo, es decir el canal 0 tiene mayor prioridad que el 1 y así sucesivamente. Si, en cambio, su valor es uno, las prioridades de los canales son rotativas. El funcionamiento de este sistema es el que sigue: inicialmente la prioridad es como si fuese fija —canal 0 más prioritario—, pero una vez que se haya atendido uno de los canales, pasa a ser el menos prioritario y todos los que tenían menos prioridad que él avanzan un puesto en cuanto a nivel de prioridad.

6.7.8 Registro de control de interrupciones

Las interrupciones aparecen cuando un canal ha completado la transferencia que le ha sido programada, siempre y cuando dichas interrupciones están permitidas. Analicemos los bits de este registro para estudiar su funcionalidad.

Autorización de interrupción. (Bits 0 al 3*).* Cada canal puede ser autorizado o no para generar una interrupción independientemente de los otros canales. Un uno en el bit correspondiente deja trascender la interrupción de fin de transmisión. Un cero, la inhibe.

Estado de interrupción de fin de transmisión. (Bit 7). Este

Registro	Dirección	CONTENIDO DEL REGISTRO Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Control de canal	1 x *	Indicador de DMA concluido	Indicador ocupado/ listo	—	—	Incremento/ decremento contador	procesador parado no parado	modo bloque robo ciclo	lectura escritura
Control de priorid.	14	Control de rotac.	—	—	—	Permiso solicitudes DMA canal 3	permiso solicitudes DMA canal 2	permiso solicitudes DMA canal 1	permiso solicitudes DMA canal 0
Control de interrupciones	15	Indicador fin ciclo DMA.	—	—	—	permiso interrupción canal 3	permiso inrterup. canal 2	permiso interup. canal 1	permiso interrup. canal 0
Encadenamiento de datos	16	—	—	—	—	Selecc. de dos ó cuatro canales	Selecc. canal encadenado bit 8	Selecc. canal encadenado bit A	Autorización encadenamiento

Tabla 6.6 Resumen del significado de los bits de control en el circuito controlador de DMA (DMAC) M6844. (*) La x se corresponde con el número de canal.

bit es un indicador de estado y sólo puede ser leído. Si su valor es uno, indica que existe una petición de interrupción al procesador. Si un canal concluye una transferencia y tiene su bit de autorización de interrupción activado, este bit también se activará y solamente se apagará cuando el microprocesador lea el registro de control de canal correspondiente al que concluyó la transferencia y causó la interrupción.

6.7.9 Registro de encadenamiento de datos

Con la función de encadenar datos pueden hacerse lecturas o escrituras de bloques de memoria de una forma repetitiva. Unicamente no podrá estar haciendo transferencias el canal «3» durante el citado encadenamiento de datos. Veamos los bits de este registro.

Autorización de encadenamiento de datos. (Bit 0). Si este bit se programa con el valor uno, queda autorizada dicha función.

Selección del canal de encadenamiento. (Bits 1 y 2). Con las cuatro combinaciones posibles de estos bits se selecciona el canal que se va a encadenar, excepción hecha de la combinación uno-uno que seleccionará el canal «3» que no está permitido.

La función de encadenamiento se lleva a cabo transfiriendo el contenido del canal «3»—dirección de memoria y número de carácteres a transmitir— dentro de los registros de dirección y contador del correspondiente canal seleccionado por los citados bits uno y dos.

Este traspaso ocurre cuando el canal que está transfiriendo a memoria concluye, es decir, cuando su registro contador de octetos llega a valer cero.

Selección de controlador de dos o cuatro canales. (Bit 3). El DMAC puede configurarse con este bit, para manejar dos o cuatro canales. Algunas señales de control se simplifican si se controlan únicamente dos canales. Si este bit vale cero, queda el M6844 como controlador de dos canales. Si vale uno, puede controlar hasta cuatro.

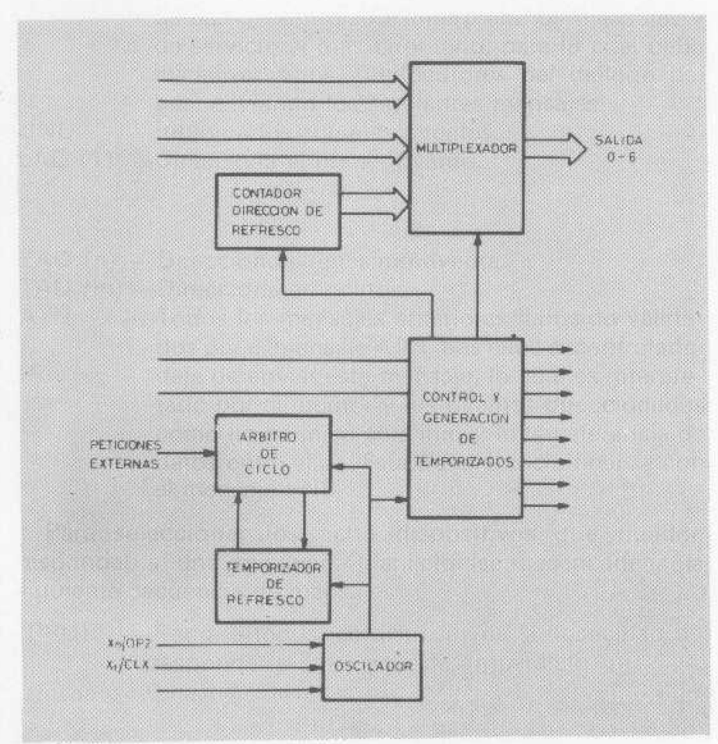


Figura 6.9 Controlador de memorias dinámicas

La tabla 6.6 resume el significado de los registros y bits de control hasta aquí descritos.

6.8 APENDICE: CONTROLADOR DE MEMORIA 8202

6.8.1 Introducción

El 8202 es un controlador de memoria dinámica diseñado para suministrar todas las señales necesarias para la utilización de memorias dinámicas en sistemas de microprocesador. Con este controlador se multiplexan las direcciones de refresco y las de lectura-escritura, así como las señales de control. También se llevan a cabo con este circuito los ciclos de refresco, bien por solicitud externa o bien por generación interna del ciclo.

6.8.2 Organización

Básicamente son seis los bloques internos del 8202: el oscilador, el árbitro de ciclo, el temporizador de refresco, el contador de dirección de refresco, el multiplexador y por último el bloque de control y generación de temporizados.

Todo esto se refleja en la figura 6.9.

6.8.3 Oscilador

El oscilador suministra los ciclos básicos para todas las funciones del circuito. Puede ser gobernado por un cristal conectado entre dos de sus patillas (X_0/OP2 y X_1/CLK).

Si no se quiere usar esta modalidad puede conectarse X_0/OP2 a 12 voltios a través de una resistencia de 1 K y aplicarse una entrada de reloj a la otra patilla.

6.8.4 Arbitro de ciclo

Los ciclos de lectura y escritura deberán hacerse cuando existe una petición externa en este sentido. Los ciclos de refresco pueden solicitarse externa o internamente. Si la petición de refresco es interna, en los casos conflictivos deberá arbitrarse el tipo de ciclo a realizar, ya que además de una demanda interna de refresco pueden llegar demandas externas aleatorias. Esta será la función que lleve a cabo el bloque «Arbitro de ciclo».

6.8.5 Temporizador de refresco

No es más que un simple temporizador que se encarga de solicitar ciclos con la periodicidad requerida. Este temporizador se pone a cero cuando se solicita un ciclo de refresco, de manera que aunque se tarde un poco en realizar dicho ciclo —por existir alguna lectura o escritura sin concluir— la petición siguiente no sufre demora y el ritmo de refrescos no se pierde.

6.8.6 Contador de refresco

Este contador contiene la dirección de la fila de memoria que debe ser refrescada. El contador de refresco es cíclico y tiene los bits necesarios para refrescar todas las filas de hasta 64 K octetos.

6.8.7 Multiplexador

A él le llegan tres grupos de direcciones que deberán ser

enviadas a los circuitos de memoria. Por un lado las direcciones que especifican la fila que debe ser refrescada en los ciclos de refresco. Por otro lado, en los ciclos de lectura o escritura, se debe enviar primero la dirección de la columna que permite seleccionar, dentro de la citada fila, una determinada celdilla de memoria. Estos dos grupos de direcciones se envían multiplexados en el tiempo y es el bloque multiplexador el que realiza esta tarea.

6.8.8 Controlador y generador de temporizados

Este bloque deberá generar las señales de mando de las memorias, según el ciclo. En un ciclo de lectura generará las siguientes señales:

1.º) Las señales de validación (strobe) de direcciones de fila
2.º) La señal de validación (strobe) de direcciones de columna.
3.º) La señal de «fin de ciclo» y validación de datos que se envía junto con los datos leídos al peticionario del ciclo de lectura.

En un ciclo de escritura las señales generadas son:

1.º) Validación de direciones de fila.
2.º) Validación de direcciones de columna
3.º) Señal de escritura de memoria
4.º) Señal de fin de ciclo.

En un ciclo de refresco se genera la señal de validación de direcciones de fila. Esta señal es suficiente en este tipo de ciclo. Por otra parte, se incrementa el contador interno de refresco, y se pone a cero el temporizador de ciclos.

convertidores A/D y D/A

7

Josep M.ª Fuertes i Armengol

7.1 INTRODUCCION

La utilización de microprocesadores como dispositivos de control de sistemas reales hace necesario un intercambio de información entre dichos sistemas y el microcomputador. Estas informaciones pueden ser señales digitales (contacto abierto o cerrado, existencia o no de luz,...) o pueden ser señales analógicas (de posición, temperatura, flujo,...) las cuales deben ser convertidas en señales eléctricas mediante transductores y posteriormente en señales digitales mediante dispositivos que realicen la conversión analógica a digital para que el microprocesador pueda realizar el algoritmo de control residente en su memoria (figura 7.1). El resultado del algoritmo puede ser un conjunto de órdenes digitales (abrir o cerrar relés, conmutar circuitos,...) o de palabras digitales que deberán acondicionarse para poder aplicarlas a elementos actuadores analógicos (posicionadores, amplificadores lineales, variadores de velocidad,...) por lo que deberán ser convertidas en señales analógicas mediante dispositivos que realicen la conversión de digital a analógico.

Se pueden distinguir en el proceso de control tres partes diferenciadas: Adquisición de datos; Realización del algoritmo; Ordenes de mando.

La figura 7.2 muestra la forma más simple de adquisición de datos: este sistema consiste en un circuito de acondicionamiento de la señal analógica (normalmente, un amplificador y un filtro.) A continuación un circuito de captura y mantenimiento (Sample and Hold en la literatura anglosajona) almacena el nivel analógico que está en un determinado instante en su entrada y lo mantiene mientras se realiza la conversión. En tercer lugar, el convertidor analógico-digital convierte el nivel analógico, en una palabra digital de n bits, la cual, una vez terminada la conversión se almacena en la memoria del microcomputador.

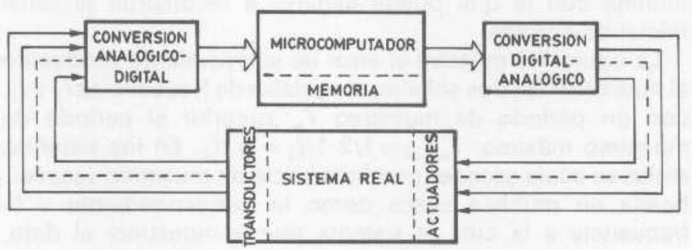


Figura 7.1 Control de un sistema real mediante microcomputador

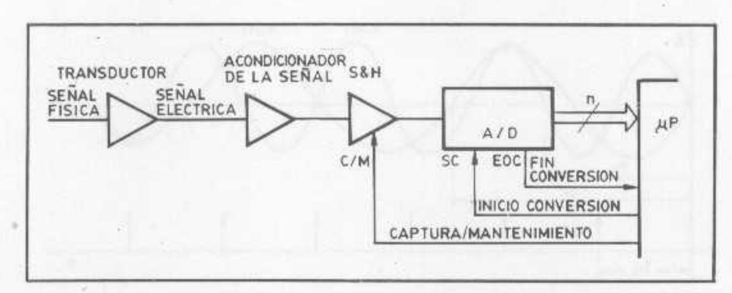


Figura 7.2 Sistema de adquisición de datos, de una única entrada analógica.

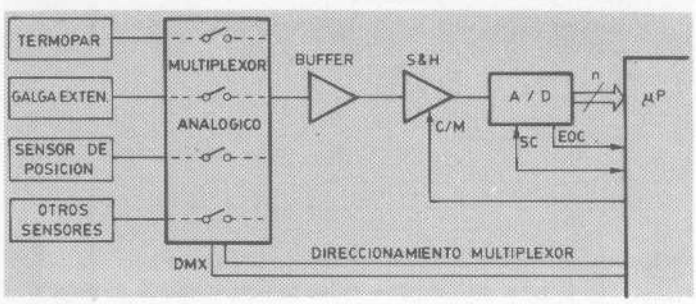


Figura 7.3 Adquisición de datos mediante multiplexor analógico

Cuando son varios los transductores que se deben leer secuencialmente, se puede utilizar un selector o multiplexor analógico entre los acondicionadores de señal y los circuitos de captura y mantenimiento (figura 7.3). El multiplexor es direccionado de tal manera que cada entrada analógica se conecta al convertidor secuencialmente. Este método permite una economía del espacio y del coste de la circuitería, ya que con un único circuito de captura y mantenimiento y un convertidor analógico-digital se pueden tratar gran cantidad de entradas analógicas, a costa de reducir la velocidad de adquisición de datos en relación a la que se obtendría utilizando tantos sistemas sencillos como entradas hubiera.

Cuando es necesario que una cierta cantidad de señales analógicas sean leídas simultáneamente, cada canal debe tener su propio circuito de captura y mantenimiento, los cuales son a continuación seleccionados secuencialmente por medio de un multiplexor para su conversión digital por medio de un convertidor analógico-digital (figura 7.4).

Las órdenes de control para mandar actuadores analógicos se obtienen a la salida del microcomputador en forma de una palabra digital. Esta se puede almacenar en un registro para mantenerla mientras se realiza la conversión D/A (figura 7.5). Cuando se desee actuar sobre varios canales utilizando

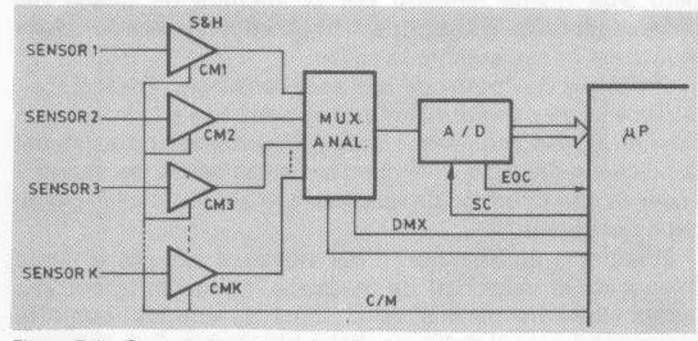


Figura 7.4 Captura simultánea de señales analógicas

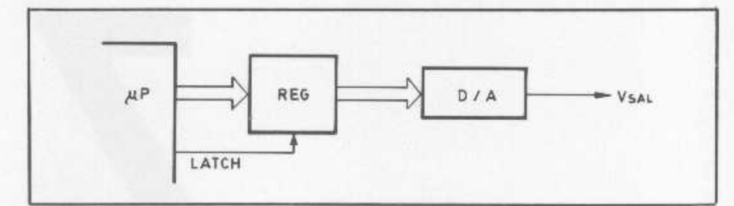


Figura 7.5 Orden de control a un canal de salida

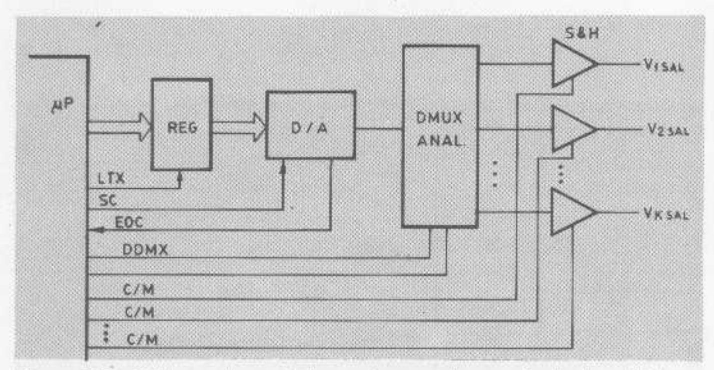


Figura 7.6 Ordenes de control a varios canales mediante demultiplexor analógico

un solo convertidor D/A, puede utilizarse un demultiplexor analógico a la salida del convertidor, seguido de tantos circuitos de captura y mantenimiento como canales de salida (figura 7.6).

7.2 PROCESO DE MUESTREO

Las señales analógicas que se pueden obtener normalmente procedentes de transductores presentan un dominio temporal y un margen de valores continuos (figura 7.7*a*). Esta misma señal muestreada, tal como muestra la figura 7.7*b*, es de valores continuos, pero el dominio temporal es discreto, o sea, en el tiempo $(t_{k+1}-t_k)$, el valor de la señal es constante. La figura 7.7*c* muestra la misma señal digitalizada, la cual presenta un dominio temporal y un margen de valores discretos; sólo puede tomar valores que se encuentran a una distancia entre ellos igual a $m(v_{i+1}-v_i)$, $m=1, 2, 3,...$

La relación entre las tres señales anteriores puede representarse según el diagrama de la figura 7.8. La señal analógica continua, $e(t)$, mediante un muestreador δ_T se convierte en una señal muestreada $\dot{e}(t)=e(t)\cdot\delta_T$, la cual a su vez se transforma en una señal digitalizada mediante un convertidor analógico-digital.

En cada uno de los procesos que sufre la señal analógica, se pierde información debido a la discretización que se realiza; se puede reducir el error en la señal muestreada haciendo el período de muestreo suficientemente pequeño, pero éste queda limitado por la anchura de banda del muestreador real, o sea por el tiempo mínimo necesario para muestrear correctamente la señal.

El análisis de Fourier de una señal analógica (descomposición en series de Fourier si ésta es periódica o la integral de Fourier en caso contrario) indica la composición espectral de dicha señal. Los coeficientes de las series de senos y cosenos dan información respecto a la amplitud y la fase de cada frecuencia.

El teorema de Shannon o del muestreo expresa el límite inferior de la velocidad de muestreo, conocida la máxima componente frecuencial de la señal de entrada, como la correspondiente al doble de dicha frecuencia. Esta frecuencia, denominada a menudo frecuencia de Nyquist, es la mínima con la que puede llegarse a reconstruir la señal inicial de entrada.

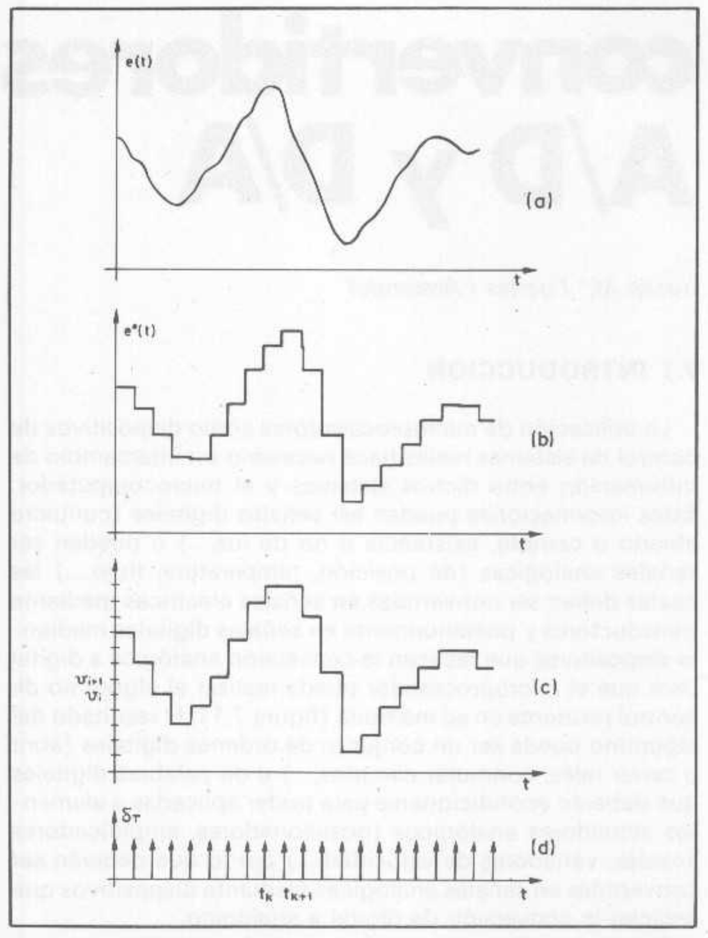


Figura 7.7 a) Señal analógica procedente de un transductor; b) la misma señal muestreada; c) idem. digitalizada; d) impulsos de muestreo.

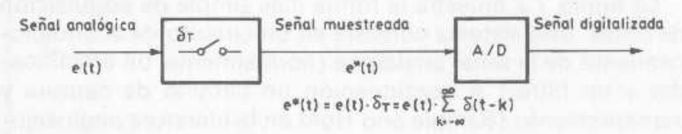


Figura 7.8 Diagrama del proceso de muestreo

La figura 7.9 muestra el error de solapamiento producido al muestrear las dos señales senoidales de frecuencias f_1 y f_2, con un período de muestreo T_m superior al período de muestreo máximo, $T_{m\,max}=1/2\cdot 1/f_2=1/2f_2$. En los sistemas reales se suele escoger una frecuencia de muestreo (especificada en muchos casos como la correspondiente a la frecuencia a la cual el sistema puede muestrear el dato, realizar la conversión y preparar el sistema para realizar una

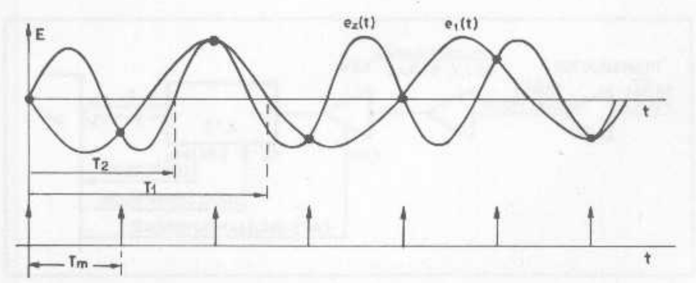


Figura 7.9 Error de solapamiento en el muestreo

nueva conversión) de un valor comprendido entre 5 y 10 veces superior a la máxima componente frecuencial significativa de la señal de entrada.

En el caso de que el sistema de adquisición no admita una frecuencia de muestreo suficientemente alta frente a una señal analógica de entrada que presente elevadas componentes frecuenciales (en muchos casos debidas al ruido superpuesto a la señal analógica) debe recurrirse a la realización de un filtrado previo de dicha señal antes de introducirla en el muestreador, el cual, aunque no elimina totamente el ruido de altas frecuencias, los puede atenuar suficientemente para que no afecten al resultado de la conversión.

En muchos casos es conveniente la realización de un filtrado digital de la señal digitalizada una vez introducida en el microcomputador.

Considérese un filtro de paso bajo RC tal como el de la figura 7.10*a*. La ecuación diferencial que lo describe es:

$$\frac{dv_0(t)}{dt} + \frac{v_0(t)}{RC} = \frac{v_i(t)}{RC}$$

La transformada de Laplace de esta ecuación es:

$$\frac{V_0(s)}{V_i(s)} = \frac{a}{s+a} = G(s)$$

donde el parámetro *a* representa la inversa de la constante de tiempo del filtro, $a = 1/\tau = 1/RC$. Si se considera una entrada de escalón unitario en el origen de tiempos, $t=0$, se obtiene en la salida del filtro:

$$V_0(s) = \frac{1}{s}\,\frac{a}{s+a}$$

La solución temporal de esta ecuación la proporciona la antitransformada de Laplace:

$$v_0(t) = 1 - e^{-at} = 1 - e^{-\frac{t}{\tau}}$$

La transformada en *z* del filtro de tiempo continuo discretizado se obtiene utilizando, por ejemplo, la aproximación de Euler: $G(s) \simeq G(s')\,|_{s'=1-z^{-1}} = H(z)$

$$H(z) = \frac{V_0(z)}{V_i(z)} = \frac{a}{1-z^{-1}+a} = \frac{m}{1-kz^{-1}}$$

donde

$$k = \frac{a}{1+a} \qquad \text{y} \qquad m = \frac{1}{1+a}$$

Reordenando la ecuación en *z*:

$$V_0(z) = k\ V_i(z) + m\ V_0(z)\cdot z^{-1}$$

Expresando la ecuación en forma de secuencia recurrente:

$$v_0(n) = k\ v_i(n) + m\ v_0(n-1)$$

El esquema funcional de dicha ecuación recurrente es el representado en la figura 7.10*b*, donde los bloques *k* y *m* representan factores multiplicativos mientras que z^{-1} se asocia a un operador de retardo unitario.

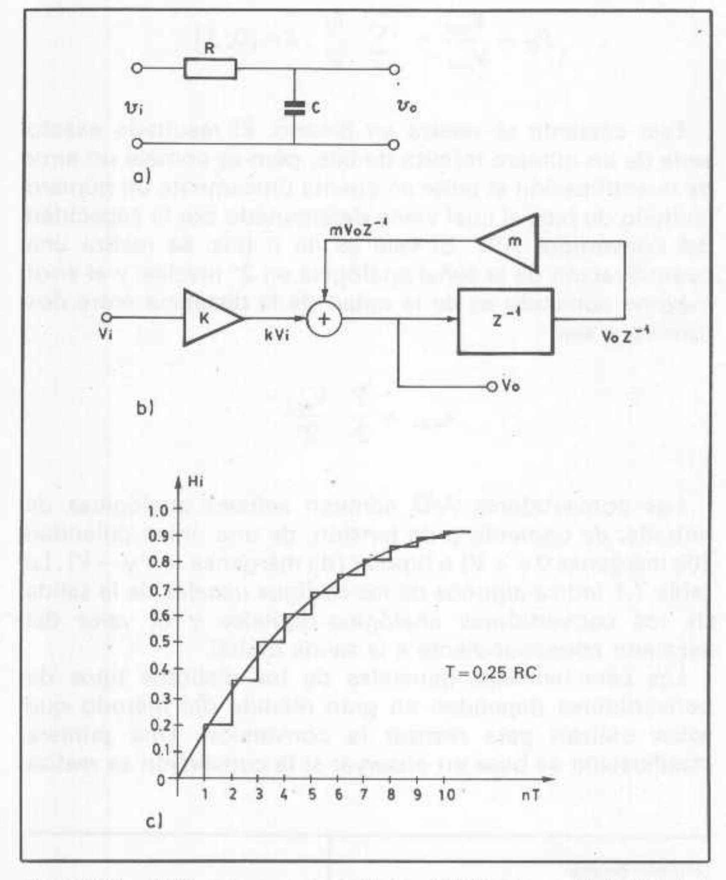


Figura 7.10 a) Filtro de paso bajo analógico; b) Filtro de paso bajo digital; c) respuesta temporal de los dos filtros

La realización de este filtro de primer orden discreto puede abordarse sin dificultades, tanto por medio de dispositivos electrónicos digitales como programándolo en un microcomputador, dado que únicamente se necesita memorizar el valor de la salida anterior y las operaciones elementales suma y producto.

La figura 7.10*c* muestra la respuesta temporal de los filtros continuo y discretizado equivalente, frente a una entrada escalón unitario en el origen de tiempos y una constante de tiempo de cuatro unidades de tiempo, realizándose la discretización con un período de muestreo *T* unitario. En estas condiciones, los parámetros *a, m,* y *k* toman por valores $a=0{,}25$, $m=0{,}8$ y $k=0{,}2$. Substituyendo estos valores en las ecuaciones continua y discretizada del filtro se observan las dos curvas de respuesta en la gráfica de la figura 7.10. Es de señalar que la aproximación entre las dos curvas muestra la equivalencia entre el circuito analógico y la configuración discreta, que será tanto más cercana cuanto menor sea el período de muestreo.

7.3 CONVERTIDORES ANALOGICO-DIGITALES

Un convertidor analógico-digital (abreviadamente C A/D) es un dispositivo que recibe una señal de entrada, V_{ent}, y la transforma en una palabra digital, P_D, con una precisión y resolución dadas, mediante una comparación con una tensión de referencia, V_{ref}. En un convertidor analógico-digital ideal, la palabra digital está relacionada con la señal de entrada por la expresión:

$$P_D = \frac{V_{ent}}{V_{ref}} = \sum_{i=1}^{\infty} \frac{\delta_i}{2^i} \; ; \; \delta = \{0, 1\}$$

Este cociente se realiza en binario. El resultado exacto sería de un número infinito de bits, pero se comete un error de cuantificación al tener en cuenta únicamente un número limitado de bits, el cual viene determinado por la capacidad del convertidor A/D. Si éste es de n bits, se realiza una cuantificación de la señal analógica en 2^n niveles, y el error máximo cometido es de la mitad de la distancia entre dos niveles, o sea,

$$\varepsilon_{max} = \frac{1}{2} \cdot \frac{V_{ref}}{2^n}$$

Los convertidores A/D admiten señales analógicas de entrada, de corriente o de tensión, de una única polaridad (de márgenes 0 y $+V$) o bipolar (de márgenes $+V$ y $-V$). La tabla 7.1 indica algunos de los códigos usuales de la salida de los convertidores analógico-digitales y el valor del escalado correspondiente a la salida digital.

Las características generales de los distintos tipos de convertidores dependen en gran medida del método que estos utilizan para realizar la conversión. Una primera clasificación se basa en observar si la conversión se realiza directamente, mediante técnicas en anillo abierto, o si se utilizan técnicas en anillo cerrado, realizándose una transformación de la entrada antes de la conversión digital. Sin ánimo de ser exhaustivos, podemos citar como métodos de conversión que han recibido una mayor atención a los siguientes:

Métodos en anillo abierto.
- Convertidor de nivel analógico a señal de frecuencia.
- Convertidor delta-sigma.
- Convertidor de codificación en cascada.
- Convertidor paralelo.

Métodos en anillo cerrado.
- Convertidor de simple rampa.
- Convertidor de doble rampa.
- Convertidor de triple rampa.
- Convertidor de aproximaciones sucesivas.
- Convertidor de seguimiento.
- Convertidor mixto de comparaciones múltiples.

El convertidor paralelo compara simultáneamente la señal analógica de entrada con una serie de niveles de referencia correspondientes a los valores del escalado cuantificados utilizando por lo tanto, tantos comparadores como niveles se desee obtener (figura 7.11). El tiempo de conversión es muy pequeño (del orden de algunos nanosegundos), pero el número elevado de componentes hace que el sistema sea costoso y, en general, de pocos bits. No se acostumbran a utilizar con microprocesadores estándar debido a la disparidad entre el tiempo de conversión y el tiempo de ciclo (de dos a tres órdenes de magnitud).

Entre los convertidores indirectos que transforman la señal de entrada en una variable temporal, los más utilizados con microprocesadores son los de simple y doble rampa y los de aproximaciones sucesivas.

El convertidor A/D de simple rampa integra una tensión de referencia hasta que la rampa generada alcanza un nivel

Unipolar binaria	
$V_s = V_{ref} \left[\sum_{i=1}^{n} \frac{\delta_i}{2^i} \right] \; ; \; \delta_i \, \{1,0\}$	11111111→9,961 V 00000000→0 V
$V_{max} = V_{ref} \left[1 - \frac{1}{2^n} \right]$	
Bipolar binaria	
$V_s = V_{ref} \left[\sum_{i=1}^{n} \left(\frac{\delta_i}{2^{i-1}} \right) - 1 \right]$	11111111→9,922 V 10000000→0 V
$V_{max\ posit.} = V_{ref} \left[1 - \frac{1}{2^{n-1}} \right]$	00000000→−10 V
$V_{max\ neg.} = -V_{ref}$	
Complemento a uno	
$V_s = V_{ref} \left[\sum_{i=2}^{n} \left(\frac{\delta_i}{2^{i-1}} \right) - \delta_1 + \frac{\delta_i}{2^{n-1}} \right]$	01111111→9,922 V
$V_{max\ posit} = V_{ref} \left(1 - \frac{1}{2^{n-1}} \right)$	00000000 11111111 } 0 V 10000000→−9,922 V
$V_{max\ neg.} = -V_{ref} \left(1 - \frac{1}{2^{n-1}} \right)$	
Complemento a dos	
$V_s = V_{ref} \left[\sum_{i=2}^{n} \left(\frac{\delta_i}{2^{i-1}} \right) - \delta_1 \right]$	01111111→9,922 V 00000000→0 V
$V_{max\ posit.} = V_{ref} \left(1 - \frac{1}{2^{n-1}} \right)$	10000000→−10 V
$V_{max\ neg} = -V_{ref}$	para 8 bits y V_{ref} = 10 V

Tabla 7.1 Códigos usuales en los convertidores A/D

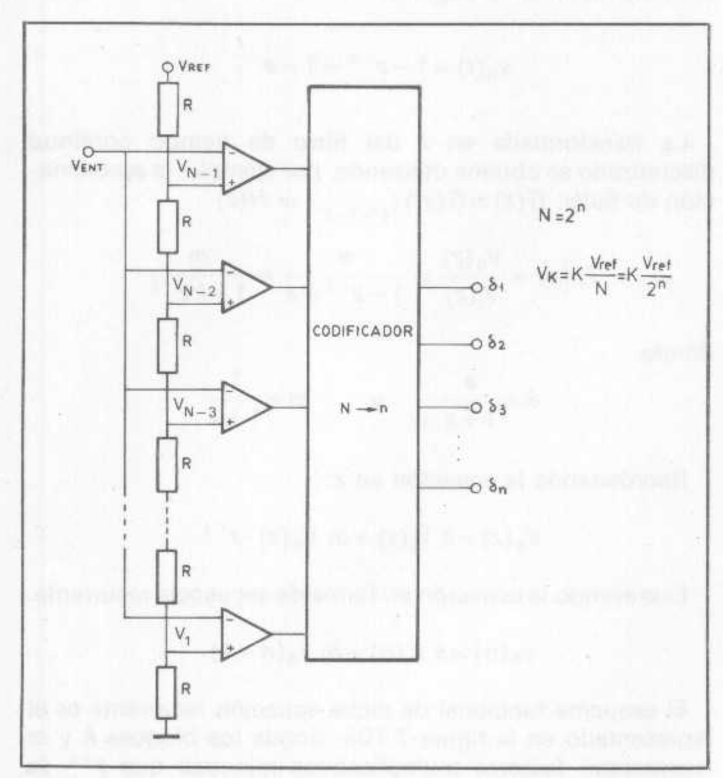


Figura 7.11 Convertidor analógico-digital paralelo

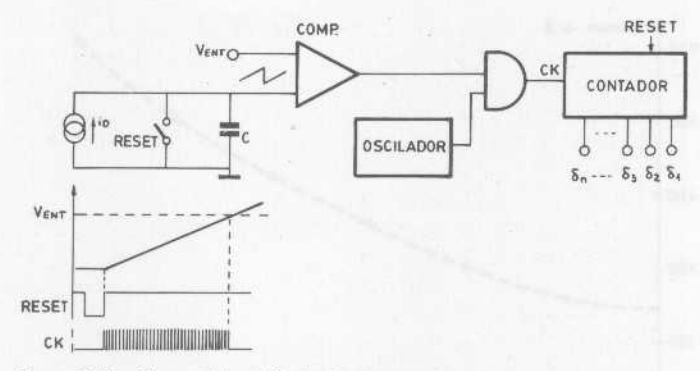


Figura 7.12 Convertidor A/D de simple rampa

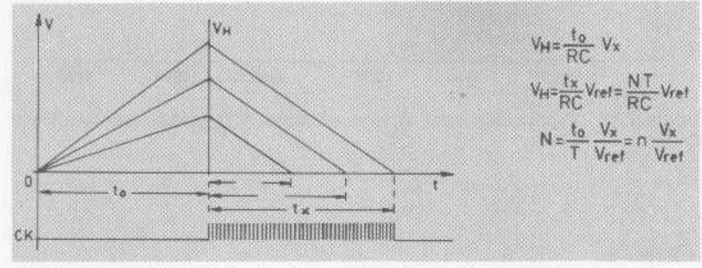


Figura 7.13 Conversión A/D de doble rampa

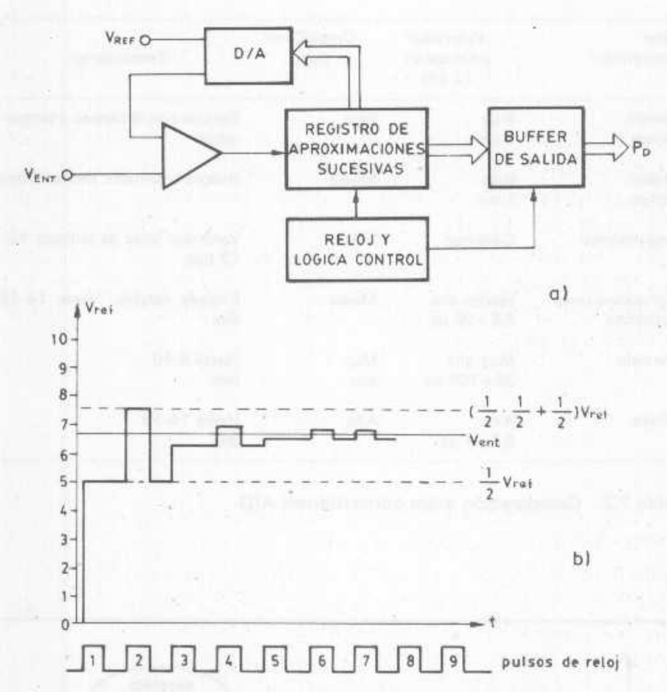


Figura 7.14 Convertidor A/D de aproximaciones sucesivas

igual a la tensión de entrada. Durante este tiempo un contador cuenta los impulsos generados por un oscilador patrón. El resultado del contador es proporcional a la tensión analógica de entrada, por lo que, escogiendo adecuadamente la frecuencia del oscilador y la tensión de referencia, se obtiene directamente la salida digitalizada (figura 7.12). El tiempo de conversión es largo y la precisión que se puede obtener no es muy buena, debido a las variaciones de la capacidad C, de la corriente I_D y de la frecuencia del oscilador a lo largo del tiempo. El *convertidor A/D de doble rampa* soluciona en gran medida los condicionamientos de precisión anteriores debido a que al realizar dos rampas consecutivas se compensan los errores producidos por las derivas de la capacidad y la frecuencia. La primera rampa (figura 7.13) se realiza integrando la entrada durante un tiempo fijo; la segunda rampa se obtiene integrando una tensión de referencia. El tiempo de integración de la segunda rampa depende del nivel que se ha conseguido durante la primera integración; durante este tiempo, un contador cuenta los impulsos de un oscilador patrón. El tiempo máximo de conversión es el necesario para contar $2 \cdot 2^n$ impulsos de reloj siendo n el número de bits de la salida digital. Supóngase que la frecuencia del reloj sea de 10 MHz, $n=8$ bits: $T_c = 2 \cdot 2^8 \cdot 10^{-7} = 50\ \mu s$.

El convertidor analógico-digital de aproximaciones sucesivas hace una optimización del tiempo de conversión mediante la realización de un algoritmo de aproximación. Utiliza un comparador, un convertidor digital-analógico y un registro de aproximaciones sucesivas (SAR, Succesive Aproximation Register) (figura 7.14). El algoritmo consiste en la aproximación bit por bit de la salida empezando por el bit de mayor peso (MSB), que corresponde a la mitad de la máxima excursión de tensión: $V_{ref} \cdot 1/2 + V_{ent}$.

Este bit valdrá 1 o 0 si el nivel de la señal de entrada está por encima o por debajo de este valor. A continuación se realiza la comparación de la tensión de entrada con una tensión que vale $\delta_1 \cdot V_{ref} \cdot 1/2 + V_{ref} \cdot 1/4$, siendo δ_1 el valor obtenido en el primer bit. Se asignará a δ_2 el valor uno o cero con el mismo criterio anterior. Se sigue realizando el algoritmo de aproximación hasta llegar al bit de menor peso (LSB). El número de iteraciones es igual al número de bits del convertidor. Este método es conceptualmente muy sencillo, permite realizar la conversión con rapidez, no exige una realización muy costosa y su tiempo de conversión es del mismo orden que el tiempo de ciclo de los microprocesadores estándar.

Existen otros convertidores analógico-digitales que por diversos motivos (coste, fiabilidad o aplicación específica) no son ampliamente utilizados. El *convertidor analógico-digital de descarga* presenta una estructura tal que la carga resultante de la entrada, se transfiere a un condensador; a continuación, éste se descarga a corriente constante al mismo tiempo que un contador acumula los impulsos procedentes de un oscilador patrón, el resultado del cual es el valor de la medida. El *convertidor de triple rampa* tiene una estructura y unas características en cuanto a fiabilidad, estabilidad y linealidad, similares al de doble rampa, pero consigue aumentar la velocidad de conversión al realizar la integración de referencia en dos etapas: en la primera etapa aproxima los bits de mayor peso y a continuación realiza la aproximación de los bits de menor peso. El *convertidor de seguimiento* (tracking) o modulador delta va resiguiendo constantemente la señal de entrada mediante incrementos o decrementos consecutivos del valor digital (modulación delta). El control del signo de los incrementos consecutivos se realiza en función de la respuesta de un comparador que tiene en sus entradas, la señal a digitalizar y la señal analógica correspondiente a la señal digitalizada, procedente de un convertidor digital-analógico que tiene incorporado. La respuesta de este convertidor es rápida siempre y cuando las variaciones de la señal analógica de entrada sean pequeñas. El *convertidor tensión-frecuencia* consiste en la conversión de la señal de tensión de entrada en una frecuencia proporcional a dicha tensión, que posteriormente se convierte en digital mediante la acumulación de los impulsos en un contador que tiene una ventana abierta durante un tiempo fijo. Es especialmente utilizado para la adquisición remota de datos y para sistemas en los que se

Tipo convertidor	Velocidad conversión 12 bits	Coste/Complejidad	Comentarios
Simple rampa	Baja 1 ms	Baja	Derivas con el tiempo y temperatura
Doble rampa	Baja 1 ms	Media	Integra la entrada. Hasta 20 bits
Seguimiento	Continuo	Baja	Variación lenta de entrada. 10-12 bits
Aproximaciones sucesivas	Media-alta 0,5 ÷ 25 μs	Media	Entrada estable. Hasta 14-16 bits
Paralelo	Muy alta 35 ÷ 100 ns	Muy alta	Hasta 8-10 bits
Mixto	Alta 0,2 ÷ 1 μs	Alta	Hasta 14-16 bits

Tabla 7.2 Comparación entre convertidores A/D

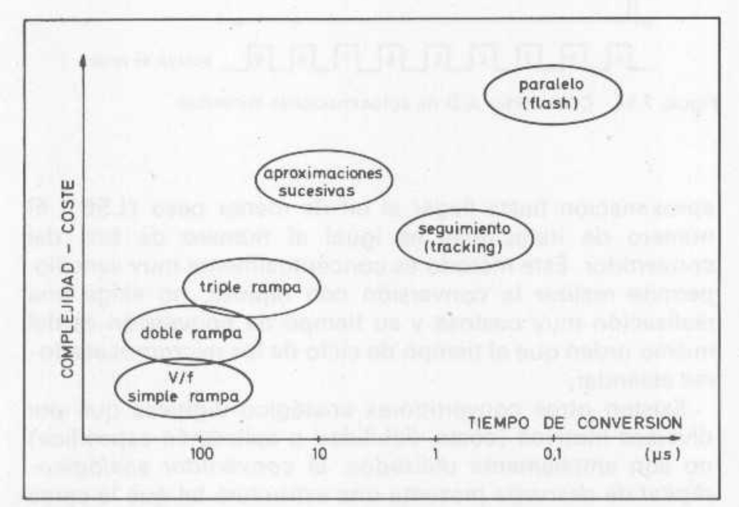


Figura 7.15 Relación entre la complejidad y el tiempo de conversión en convertidores analógico-digitales.

desea aislar la sección analógica de entrada y la sección digital, pues sólo es necesario una señal de comunicación entre las partes analógica (remota) y la digital.

En general, se puede indicar que cuanto menor sea el tiempo de conversión, mayor será la complejidad y, a su vez, el coste de los diversos tipos de convertidores A/D (tabla 7.2, figura 7.15). Esta relación coste-prestaciones condiciona que los convertidores más utilizados sean:

Los *convertidores de doble rampa* o doble integración, que aun siendo los más lentos, ofrecen alta resolución, precisión e inmunidad al ruido a unos precios moderados y son usados en aplicaciones de control de procesos e instrumentación. Los *convertidores paralelo*, que por su rapidez se usan en aplicaciones de tratamiento de imagen y otras aplicaciones de alta velocidad, de resolución limitada a 8 o 10 bits y con un coste relativamente elevado, aunque se espera que su índice de producción futuro sea elevado. Los *de aproximaciones sucesivas*, con el mayor índice de producción, aúnan unas buenas prestaciones para aplicaciones generales con un coste moderado y una buena resolución. Los *convertidores mixtos* (subranging) utilizan unas técnicas combinación de las dos anteriores y cubren la zona intermedia de los dos convertidores anteriores en cuanto a velocidad de conversión, resolución y coste.

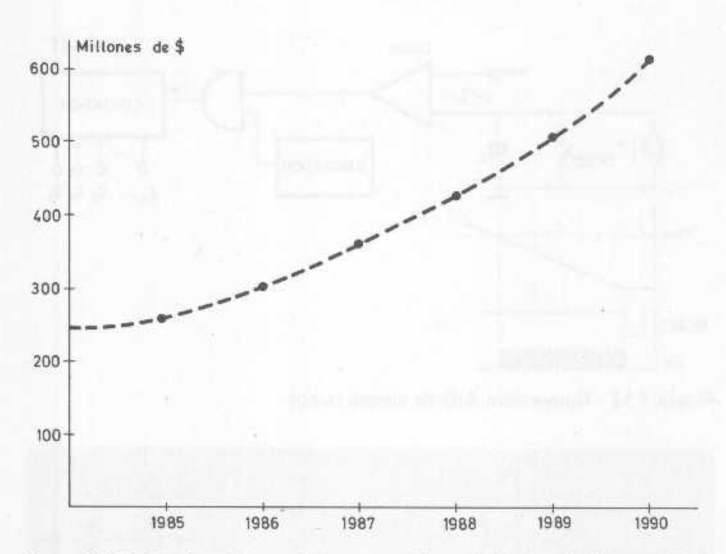


Figura 7.16 Producción total de convertidores A/D (según Venture Dev. Co.)

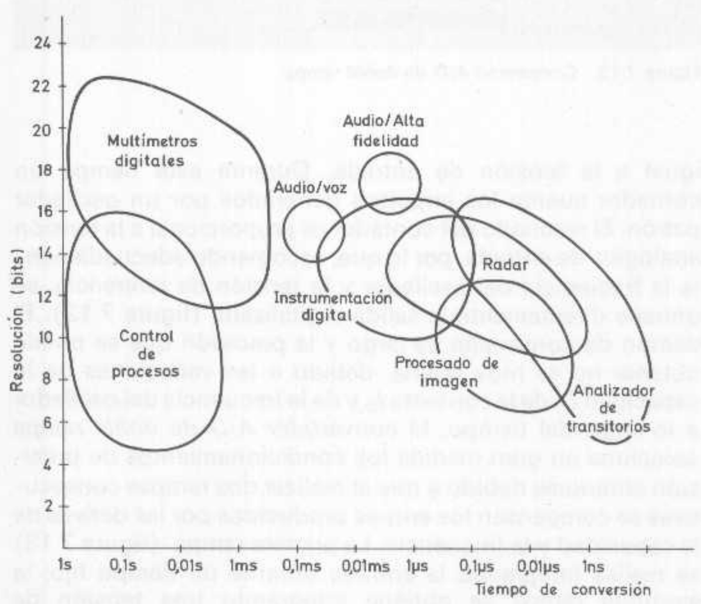


Figura 7.17 Aplicación de los convertidores A/D

En conjunto, se espera que el crecimiento de producción de los convertidores A/D hasta el año 1990 será superior al 25 % anual (fig. 7.16).

7.4 CONVERTIDORES DIGITAL-ANALOGICO

Un convertidor digital-analógico (abreviadamente, convertidor D/A) es un dispositivo que recibe una información digital en forma de una palabra de *n* bits, y la transforma en una señal analógica. La transformación se realiza mediante una correspondencia entre 2^n combinaciones binarias posibles en la entrada y 2^n tensiones (o corrientes) discretas obtenidas a partir de la tensión de referencia V_{ref}. La señal analógica así obtenida no es una señal continua, sino que se obtiene un número discreto de escalones a consecuencia de la discretización de la entrada. El proceso consiste en hacer una ponderación de cada bit y sumar el resultado de todas las ponderaciones multiplicadas por su valor (0 o 1).

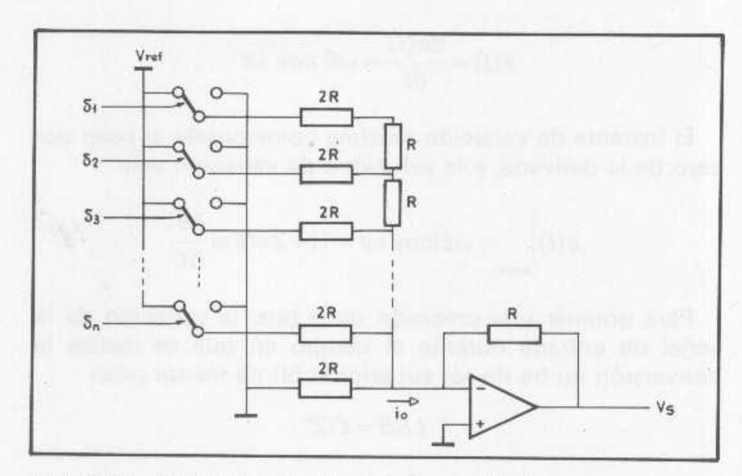


Figura 7.18 Convertidor D/A de resistencias ponderadas

$$V_{SAL} = \sum_{i=1}^{n} V_{pond.i} \cdot \delta_i; \ \delta = \{0, 1\}$$

La ecuación anterior sugiere una realización inmediata: el *convertidor digital-analógico de resistencias ponderadas.* Este convertidor utiliza un sumador de n canales, siendo n el número de bits, seleccionándose cada canal o no, según el valor del bit correspondiente (0 o 1). En el caso de un convertidor lineal, el valor de la resistencia de cada canal $R_i = 2^i \cdot R$ (figura 7.18)

$$V_{SAL} = -R \sum_{i=1}^{n} \delta_i \cdot \frac{V_{ref}}{2^i R} = -V_{ref} \sum_{i=1}^{n} \delta_i \cdot \frac{1}{2^i}$$

Este circuito tiene el inconveniente de que necesita resistencias de gran precisión de valores distintos. Al mismo tiempo, los conmutadores de canal deben tener unos valores de resistencia de conducción y de bloqueo en unos márgenes suficientemente alejados de los valores de las resistencias del circuito para que no interfieran en la conversión. Por ejemplo, si se toma $R = 5$ K y $n = 12$ bits, el valor de las resistencias varían desde $R = 5$ K hasta $R_{12} = 2^{12} \cdot R = 20{,}48$ MΩ.

El convertidor digital-analógico por red de resistencias R-2R, soluciona los problemas expuestos anteriormente al utilizar resistencias con sólo dos valores; la corriente que circula por las resistencias $2R$ (figura 7.19), está en progresión geométrica de razón 1/2 debido a que en cada punto de unión de las resistencias $R-R$, la impedancia del circuito es igualmente R. Seleccionando cada canal en función del bit correspondiente (figura 7.20), se obtiene una corriente a la entrada del amplificador operacional suma de las corrientes de cada rama, la cual produce una tensión analógica de salida V_{sal}:

$$i_0 = \sum_{i=1}^{n} \delta_i \ \frac{I}{2^i}$$

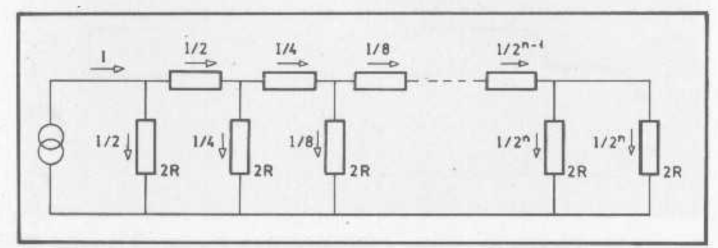


Figura 7.19 Red resistiva R-2R

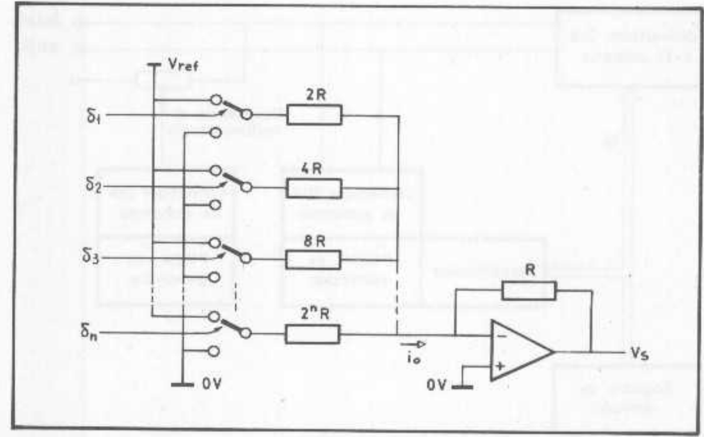


Figura 7.20 Convertidor D/A de red resistiva R-2R

$$V_{SAL} = -i_0 R = -RI \sum_{i=1}^{n} \delta_i \ \frac{1}{2^i} = -V_{ref} \sum_{i=1}^{n} \delta_i \cdot \frac{1}{2^i}$$

El *convertidor digital-analógico por modulación de anchura de impulso* es un convertidor que en una primera etapa convierte el código digital en una señal pulsante de anchura controlada y en una segunda etapa y mediante un filtro de paso bajo, convierte esta señal pulsante en una señal analógica. En la figura 7.21 puede observarse la estructura de este convertidor: Un comparador de n bits compara el código digital con el valor de un contador de n bits que se va incrementando con los impulsos de un oscilador patrón. Al alcanzar éste el valor del código digital, su salida bascula. El período de la señal modulada es por lo tanto, el tiempo necesario para contar 2^n impulsos, mientras que la anchura de la señal modulada es igual al tiempo necesario para contar tantos impulsos como indica la palabra digital de entrada. El filtrado de salida implica una velocidad de conversión lenta, pero por contra este convertidor es estructuralmente monotónico.

En aplicaciones en las que se desea obtener unas prestaciones elevadas, principalmente en precisión, resolución, tiempos pequeños de conversión y monotonicidad se requiere otro tipo de solución. Una de las soluciones utilizadas para mejorar las prestaciones de los convertidores

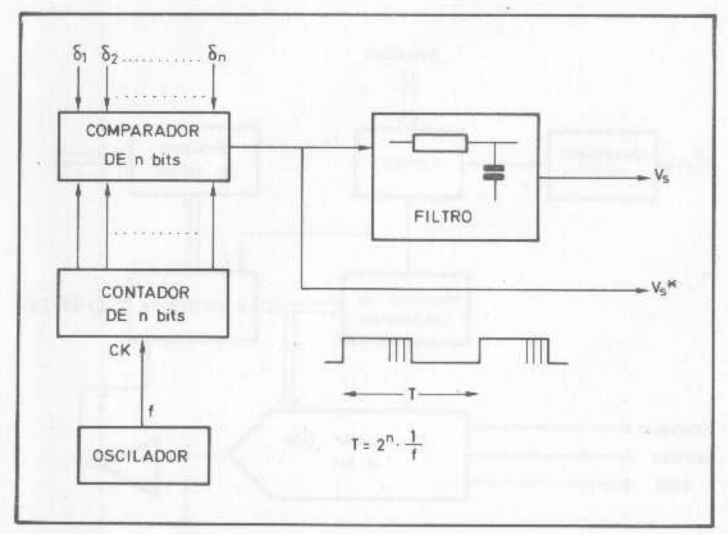


Figura 7.21 Convertidor digital-analógico de modulación de anchura de impulso

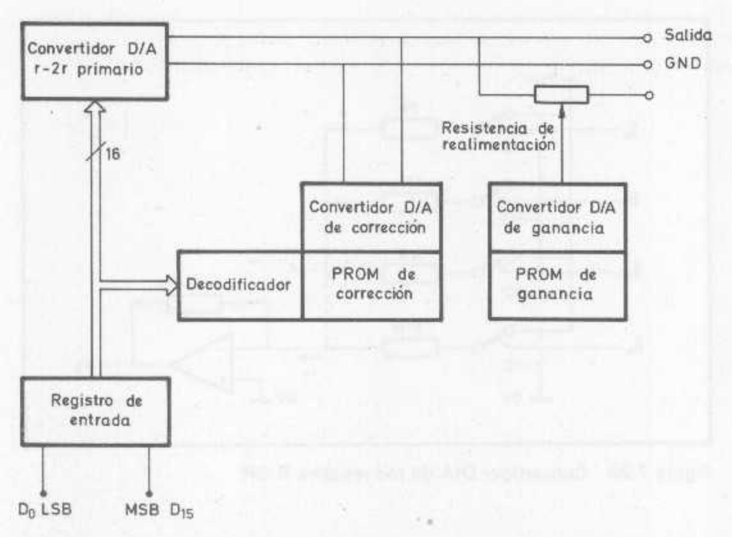


Figura 7.22 Corrección mediante PROM de un convertidor D/A (ICL7121) (Electronic Design, diciembre 1986)

D/A consiste en la inclusión de memorias PROM en la pastilla del convertidor con la finalidad de superponer la corrección de monotonicidad y ganancia en la etapa de salida (figura 7.22). En los convertidores A/D se mejoran las prestaciones (figura 7.23) utilizando complejos microcontroladores que regulan las operaciones internas del funcionamiento del proceso de aproximaciones sucesivas, la captura y el muestreo de las señales de entrada y controla los procedimientos de ajuste y autocalibrado en línea del convertidor.

7.5 CIRCUITOS DE CAPTURA Y MANTENIMIENTO

Supóngase que se desea realizar una conversión A/D de una señal senoidal $e(t)$, de frecuencia f y se desea una resolución de n bits.

$$e(t)=E \text{ sen } \omega t=E \text{ sen } 2\pi ft$$

La velocidad de variación de dicha señal viene indicada por su primera derivada

$$\dot{e}(t)=\frac{de(t)}{dt}=\omega E \cos \omega t$$

El instante de variación máxima corresponde al paso por cero de la derivada, y la velocidad de variación vale

$$\dot{e}(t)\bigg|_{max}=\omega E|\cos \omega t=1|=2\pi fE\simeq\frac{\Delta e}{\Delta t}$$

Para obtener una precisión de n bits, la variación de la señal de entrada durante el tiempo en que se realiza la conversión no ha de ser superior al bit de menor peso

$$1\ LSB=E/2^n$$

$$\Delta e\leqslant\frac{E}{2^n}=2\pi f_{max}E\Delta t$$

De este resultado se desprende que la frecuencia máxima de la señal que se podrá convertir con n bits de precisión es

$$f_{max}=\frac{1}{2\cdot\pi\cdot 2^n\cdot\Delta T}$$

siendo ΔT el tiempo de apertura del convertidor.

Si se desea utilizar un convertidor de 8 bits, con un tiempo de conversión (apertura efectiva) de 30 μs.

$$f_{max}=\frac{1}{2\cdot\pi\cdot 2^8\cdot 30\cdot 10^{-6}}=20{,}8\text{ Hz}$$

Este resultado indica la conveniencia de utilizar circuitos de captura y mantenimiento, ya que las frecuencias máximas para este tipo de conversión son muy pequeñas.

Los *circuitos de captura y mantenimiento* (Sample and Hold) son elementos analógicos equivalentes a los registros capturadores (latch) digitales. Se utilizan para muestrear una señal analógica y mantener el nivel de dicha señal en un medio de almacenamiento (normalmente un condensador) para proceder a su medida o a su procesado en un sistema.

El funcionamiento de un circuito de captura y mantenimiento ideal muestra que éste sigue la entrada durante el período de muestreo y luego mantiene el último valor analógico de tensión, cuando conmuta al modo de mantenimiento. La figura 7.24 muestra un circuito esquemático de un S & H y la señal que se obtiene a su entrada y salida

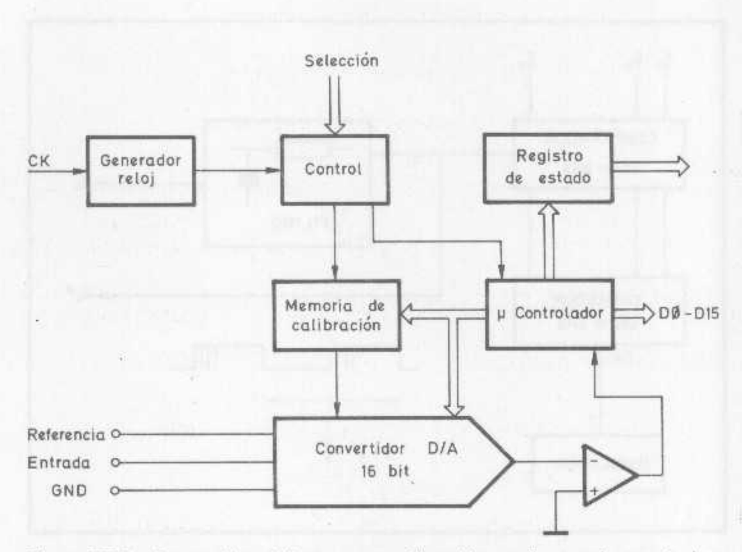


Figura 7.23 Convertidor A/D con autocalibración mediante microcontrolador (CS5016) (EDN, enero 1987)

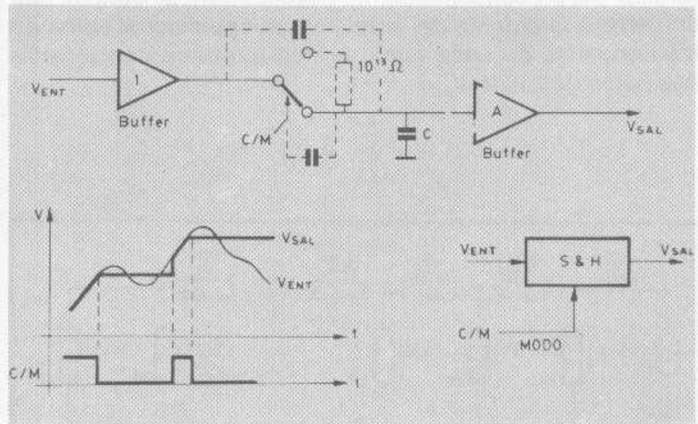


Figura 7.24 Circuito de captura y mantenimiento: Formas de la señal de entrada y salida de un circuito ideal

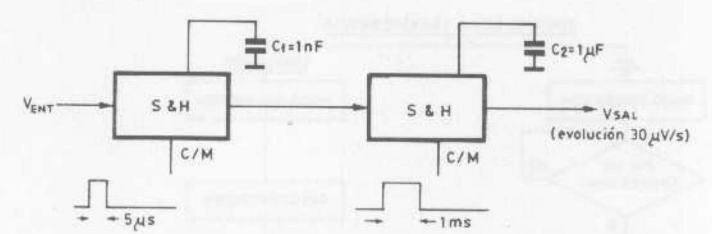


Figura 7.25 Circuito S & H de captura rápida y mantenimiento prolongado

durante su funcionamiento normal. Tanto el tiempo de captura como el tiempo de mantenimiento dependen del valor del condensador de almacenamiento.

Si se desea obtener un circuito S & H que tenga un tiempo de captura muy pequeño deberá utilizarse un condensador de baja capacidad, mientras que si se desea que la señal capturada se mantenga durante un largo tiempo, deberá utilizarse un condensador de gran capacidad. Utilizando dos circuitos S & H tal como indica la figura 7.25, pueden satisfacerse a la vez las dos condiciones anteriores.

7.5.1 Multiplexores analógicos

Un multiplexor analógico es un dispositivo que tiene un conjunto de entradas, una salida y una serie de conmutadores analógicos direccionables que van seleccionando aleatoriamente o secuencialmente cada una de las entradas y las conecta a la salida. En cada instante sólo se cierra un conmutador y la secuencia de muestreo se realiza mediante un circuito de control o por el microcomputador directamente.

Un multiplexor ideal presenta un tiempo de conmutación de cada canal y una impedancia de paso nulos, no introduce distorsión en la señal y cuando un canal está abierto, hay aislamiento completo entre su entrada y la salida. Los conmutadores analógicos están basados en dispositivos de efecto de campo. La tabla 7.3 indica algunas de las características de los distintos transistores de efecto de campo.

Los multiplexores reales presentan una capacidad parásita de entrada, C_E (figura 7.26), que contribuye a limitar la anchura de banda de funcionamiento. Cuando un conmutador está abierto, presenta una impedancia de paso muy elevada (del orden de 10^{13} Ω) mientras que cuando está cerrado, es de algunos pocos ohmios (de 30 a 150 Ω típicamente).

Según sea la configuración utilizada, los multiplexores pueden conmutar señales unipolares o diferenciales. El modo unipolar (figura 7.27) se utiliza en aplicaciones en las

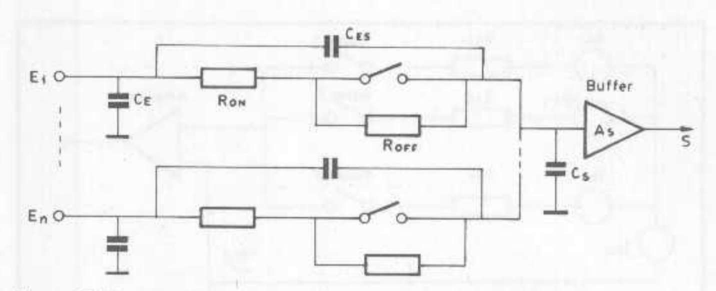


Figura 7.26 Circuito equivalente de un multiplexor analógico

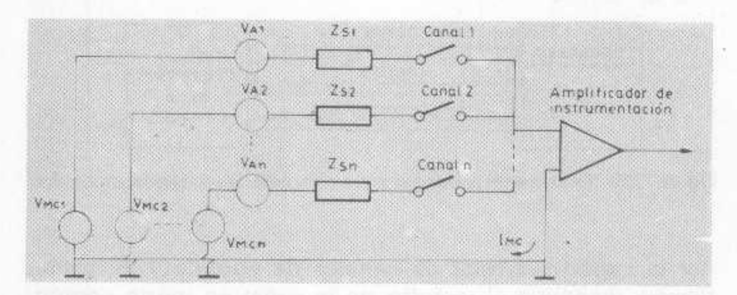


Figura 7.27 Multiplexor analógico en modo unipolar

que el nivel de señal es significativamente superior a las tensiones en modo común presentes en el sistema. Cuando las medidas son de bajo nivel y puede sumarse ruido en modo común, la entrada diferencial es la más adecuada (figura 7.28), debido a la utilización de la capacidad de rechazo en modo común (CMR) del amplificador, aunque se reduce a la mitad la capacidad de canales de entrada. La configuración cuasidiferencial (figura 7.29) permite mante-

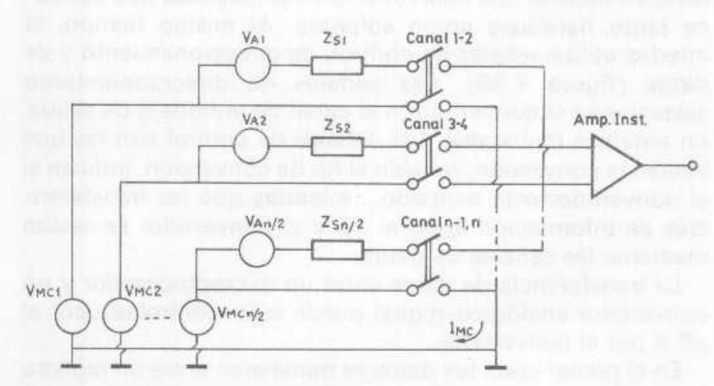


Figura 7.28 Multiplexor analógico en modo diferencial

Tipo de interruptor	V_{DS} *(ON)* Ω	*Margen de tensión analógica*	I_D *(off)* nA	t_{on} ns	t_{off} ns	*Corriente alimentación*
PMOS	75 a 800	± 10 V	1 a 4	200 a 300	400 a 2000	0,001 a 3 mA
CMOS	25 a 500	± 15 V	0,02 a 5	20 a 1000	20 a 500	0,0005 a 4 mA
N-JFET	10 a 100	+10 a −15 V +15 a −10 V	0,1 a 15	150 a 750	130 a 1250	0,001 a 4 mA
P-JFET	100 a 150	+20 a 0 V	0,2 a 0,5	100 a 500	100 a 500	
Bi-FET/JFET	60 a 200	±11 V	0,1 a 3	350 a 500	90 a 300	4,5 a 10 mA

Tabla 7.3 Características de los transistores de efecto de campo

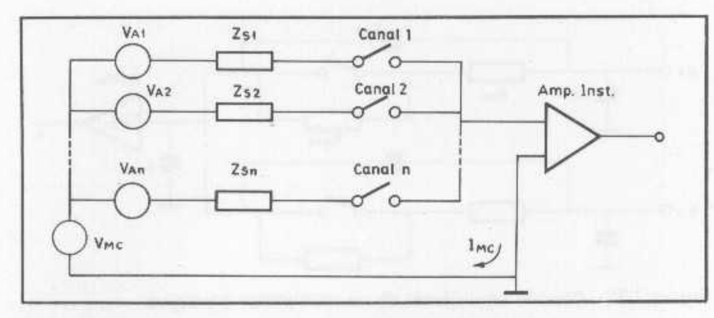


Figura 7.29 Multiplexor analógico en modo cuasidiferencial

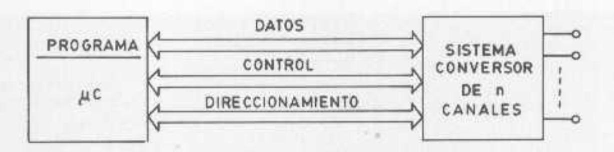


Figura 7.30 Esquema de la interfaz entre convertidores y microcomputador

ner la capacidad total de canales de entrada y al mismo tiempo mantiene el rechazo de la señal en modo común, pero sólo se puede utilizar en los casos en que las señales se obtienen en unas condiciones de ruido comunes.

La utilización de un amplificador de instrumentación de alta impedancia de entrada (superior a 10^8 ohmios) colocado en la salida del multiplexor permite minimizar los errores debidos a las impedancias finitas de conducción de los canales del selector analógico.

7.6 INTERFAZ ENTRE CONVERTIDORES Y μC

La interfaz entre un convertidor y un microcomputador se realiza mediante una interconexión multifacética que contiene tanto hardware como software. Al mismo tiempo, la interfaz utiliza señales de control, de direccionamiento y de datos (figura 7.30). Las señales de direccionamiento seleccionan el convertidor o el canal de entrada o de salida, en sistemas multicanal. Las señales de control son las que inician la conversión, señalan el fin de conversión, indican si el convertidor está ocupado... mientras que las transferencias de información entre el μC y el convertidor se realiza mediante las señales de datos.

La transferencia de datos entre un microprocesador y un convertidor analógico-digital puede estar controlada por el μP o por el convertidor.

En el primer caso, los datos se transfieren entre un registro asociado al convertidor y el registro acumulador u otro registro interno del μP, según se utilice una instrucción de E/S o la transferencia sea mapeada en memoria (Memory Mapped). En uno y otro caso, el microprocesador debe esperar, bien sea mediante un bucle de espera o en estado de WAIT, a que el convertidor acabe la conversión para entrar el dato válido (figura 7.31).

Cuando es el convertidor A/D el que inicia la transferencia de datos, se utilizan los métodos de transferencia por interrupción, por acceso directo a memoria (Direct Memory Access, DMA) o mediante memoria auxiliar.

Transferencia por interrupción: El μC selecciona el canal de entrada, inicia el proceso de conversión y luego prosigue con su tarea. Al acabarse la conversión, la señal *EOC* (End of Conversion) interrumpe al μP. Si la interrupción es vectorizada, el convertidor envía la dirección de la rutina de servicio, mientras que si no es vectorizada, el μP debe

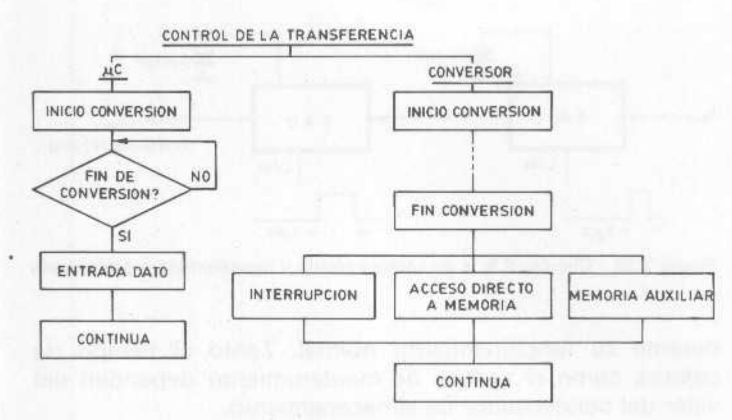


Figura 7.31 Transferencia entre un convertidor A/D y un μC

examinar todos los convertidores hasta saber cuál ha sido el que la ha enviado, para poder así servirlo. Esta transferencia no necesita una circuitería compleja, es útil cuando los convertidores presentan un tiempo de conversión elevado, pero el programa interruptivo debe estar alojado en la memoria.

La transferencia por acceso directo a memoria permite al convertidor un funcionamiento continuo. Después de cada conversión, la señal *EOC* inicia la transferencia de la palabra convertida a la memoria principal. Pero sólo puede realizarse la transferencia durante el tiempo en que el bus de datos y de direcciones del microcomputador no está utilizado, o sea, provocando el estado de retención (Hold) del μP, o por medio del «robo del ciclo». Este método es especialmente útil cuando hay gran cantidad de convertidores; se necesita un controlador de bus, aumentándose la complejidad de la circuitería necesaria, pero no se pierde tiempo de procesador ni se necesita programa de gestión de la conversión, puesto que el conversor está en funcionamiento continuo o sea que al acabar una conversión y después de almacenarla en la memoria, vuelve a autoiniciarse la conversión.

La transferencia mediante memoria auxiliar es similar al caso anterior, pero los datos obtenidos se guardan en una memoria auxiliar y el procesador los lee de allí. Se introducen retrasos (incluso de un período de iteración) debido a que las señales que se memorizan son muestreadas sólo una vez en cada período de iteración.

La transferencia entre un μC y un convertidor digital-analógico se inicia mediante el envío de la palabra o palabras digitales (en caso de que el convertidor sea de mayor número de bits que el bus del μC), al registro del convertidor. Puede utilizarse también la estructura de mapeado de memoria ya que el registro se comportará como una (o varias) posiciones de memoria de sólo escritura, permitiendo así una programación sencilla y más potente, pues se podrán utilizar todas las instrucciones de referencia a memoria para realizar la transferencia al convertidor.

Según sea la estructura utilizada, habrá un mayor peso en la componente de hardware o en la de software. Son de uso bastante común las tarjetas preparadas para interconectar en el bus estándar de algunos sistemas microcomputador, que incluyen un sistema de adquisición de datos, un conjunto de convertidores y la parte que asegura la compatibilidad entre las alimentaciones y niveles lógicos de las señales digitales que se intercambian. Los convertidores discretos directamente acoplables al bus de los μP acostumbran a utilizar el estado de alta impedancia en el registro de salida. Pueden ser monolíticos o híbridos, consiguiéndose en los primeros una gran flexibilidad de aplicación, mientras que los híbridos

Compatibilidad bus-µC/ bits	Fabricante	Referencia	Tiempo de conversión	A/D bits	Canal	D/A bits	Técnica de conversión	Tecnología	n.º patillas (pin)	Comentarios
			Especificaciones							
General 8 bits	Analog Devices	AD570	20 µs	8	1	—	Aprox. Suc.	CMOS	16 pin DIP	
		AD571	25 µs	10	1	—	Aprox. Suc.	CMOS	18 pin DIP	
		AD574	25 µs	12	1	—	Aprox. Suc.	CMOS	28 pin DIP	
		AD363	25 µs/can.	12	16	—	Aprox. Suc.	Bipolar	32 pin DIP	8 canales diferenciales
		AD364	25 µs/can.	12	16	—	Aprox. Suc.		32 pin DIP	8 canales diferenciales
		AD670		8	1	—	Delay line	IIL	20 pin DIP	Apro. Suc. asíncrono
		AD7524	15 µs	8	1	—	Aprox. Suc.	CMOS	16 pin DIP	Interfase como RAM o ROM lenta
		AD7572	5 µs	12	1	—	Aprox. Suc.	CMOS/ Bip.		
		AD7578	100 µs	12	1	—	Aprox. Suc.	CMOS		
		AD7582	100 µs/can	12	4	—	Aprox. Suc.	CMOS		
		AD1170	1 ms	12	1	—	Rampa	CMOS	Modular	Incorpora un microcontrolador
General 8-12-16 bits	Burr-Brown	MP20/21	40-200 µs/can	8	16	—	Aprox. Suc.	Híbrido	80 pin Quad	Directo al bus µC
		MP22	45 µs/can	12	16	—	Aprox. Suc.	Híbrido	80 pin Quad	Directo al bus µC
		ADC574	25 µs	12	1	—	Aprox. Suc.	CMOS	28 pin DIP	
		ADC674	15 µs	12	1	—	Aprox. Suc.	CMOS	28 pin DIP	
		ADC80MAH12	25 µs	12	1	—	Aprox. Suc.		32 pin DIP	Salida paralelo y serie
General 8-16 bits	Cristall Semiconduc.	CS5012/16	16/20 µs	12/16	1	—	Aprox. Suc.	CMOS	40 pin	Con microcontrolador de autocalibrado y T&H
		CS5212	1 µs	12	1	—	Mixta	CMOS		
		CS5316	62,5 µs	16	1	—	Delta-Sigma	CMOS	18 pin DIP	con T & H
General	Datel	ADC HC12	300 µs	12	1	—	Aprox. Suc.	CMOS	32 pin DIP	
		ADC500	0,5 µs	12	1	—			32 pin	Híbrido
General	Ferranti	ZN433	1 µs/bit	10	1	—	Seguimiento			
		ZN437	20 µs/can	8	8	—	Aprox. Suc.	CMOS	28 pin DIP	Con RAM de 8×8 bits
		ZN439	5 µs	8	1	—	Aprox. Suc.	CMOS	22 pin DIP	
		ZN447/8/9	9 µs	8	1	—	Aprox. Suc.	CMOS	18 pin DIP	
General 8-12-16 bits	Harris	HI774	8,5 µs	12	1	—	Aprox. Suc.	CMOS	28 pin DIP	
		HI774A	7 µs	12	1	—	Aprox. Suc.	CMOS	28 pin DIP	
		HY9574	25 µs	12	1	—	Aprox. Suc.	CMOS	32 pin DIP	Con S&H
		HY9674	15 µs	12	1	—	Aprox. Suc.	CMOS	32 pin DIP	Con S&H
		HY94741/2	23 µs	12	1	—	Aprox. Suc.	CMOS	28 pin DIP	Con T&H
General	Hibrid Sys.	HS9476	21 µs	14	1	—			32 pin	Híbrido con S&H
		HS9516	100 µs	16	1	—				Híbrido
General 4-8-12 bits	Honeywell	HADC574Z	25 µs	12	1	—	Aprox. Suc.	BEMOS	28 pin DIP	con S&H
		HADC674Z	15 µs	12	1	—	Aprox. Suc.	BEMOS	28 pin DIP	con S&H
		HADC77100	60 ns	8	1	—	Paralelo	Bipolar	42 pin DIP	compatible con SONY CX20116
General	Intersil	8052/	3-30 Hz	10 a 16	1	—	doble rampa	PMOS	14-40 pin	Pares lentos pero precisos
		7101/03/04						CMOS		
		7109	0,1-15 Hz	12	1	—	doble rampa	CMOS	40 pin	12 bit más signo y overflow
		ILC7115	40 µs	14	1	—	Aprox. Suc.	CMOS		
General	Micro Linear	ML2200	25 µs/can	13	4 dif.	—	Aprox. Suc.	CMOS		Configurado como un subsistema de adquisición de datos
General	Micro Networks	MN5500	25 µs	12	1	—	Aprox. Suc.		40 pin DIP	
		MN5150	2,5 µs	8	1	—	Aprox. Suc.		24 pin DIP	Mapeado en memoria
		MN7120	12 µs/can	8	8	—	Aprox. Suc.		32 pin DIP	
General	Motorola	MC14442	32 µs/can	8	11	—	Aprox. Suc.	CMOS	28 pin DIP	Con S&H
		MC14433	300 µs/can	8	1	—	Simple rampa	CMOS	18 pin DIP	Con S&H
		MC14444	32 µs/can	8	15	—	Aprox. Suc.	CMOS	28 pin DIP	Con S&H
		MC145040/1	32 µs/can	8	11	—	Aprox. Suc.	CMOS	28 pin DIP	Comunicación serie

Tabla 7.4 Características de algunos convertidores A/D y D/A.

			Especificaciones							
Compatibilidad bus-μC/bits	Fabricante	Referencia	Tiempo de conversión	A/D bits	Canal	D/A bits	Técnica de conversión	Tecnología	n.º patillas (pin)	Comentarios
8080	National	ADC0808/09	100 μs/can	8	8	—	Aprox. Suc.	CMOS	28 pin DIP	Mapeado en memoria
		ADC0816/17	100 μs/can	8	16	—	Aprox. Suc.	CMOS	40 pin DIP	Mapeado en memoria
COP, 8048		ADC830	30 μs/can	8	2	—	Aprox. Suc.	CMOS	8 pin DIP	Varios canales simples o la mitad diferenciales, con comunicación serie.
		ADC831	30 μs	8	1 dif.	—	Aprox. Suc.	CMOS	8 pin DIP	
		ADC833	30 μs/can	8	4	—	Aprox. Suc.	CMOS	14 pin DIP	
		ADC834	30 μs/can	8	2 dif.	—	Aprox. Suc.	CMOS	14 pin DIP	
		ADC837	30 μs/can	8	8	—	Aprox. Suc.	CMOS	20 pin DIP	
		ADC838	30 μs/can	8	4 dif.	—	Aprox. Suc.	CMOS	20 pin DIP	
μC 8 bits		ADC0811	32 μs	8	1	—	Aprox. Suc.	CMOS	28 pin DIP	
		ADC0829	256 μs	8	1	—	Aprox. Suc.	CMOS	28 pin DIP	
		ADC0841	40 μs	8	1	—	Aprox. Suc.	CMOS	20 pin DIP	
		ADC0844	40 μs/can	8	4	—	Aprox. Suc.	CMOS	20 pin DIP	
		ADC1205	100 μs	13	8	—	Mixta	CMOS	24 pin DIP	12 bit+signo. ROM de compensación
μC 16 bits		ADC12025	100 μs	13	8	—	Mixta	CMOS	28 pin DIP	
General	RCA	1860	3,2 MHz 5 V 6,4 MHz 10 V	8				CMOS	18 pin DIP	Frecuencia de conversión, función de alimentación
		CA3010	6,5 μs	10	1	—		CMOS		Redistribución de carga
General	Signetics	ADC0801/05	25 μs	8	1 dif.	—	Aprox. Suc.	CMOS	20 pin DIP	
		5028	40 μs	8	1	—	Aprox. Suc.	IIL	24 pin DIP	
		SE/NE5030	4 μs	10	1	—	Mixto	Bipolar		
General	Texas Ins.	TLC532A	15 μs/can	8	11	—	Aprox. Suc.	CMOS	28 pin DIP	S&H, 6 Ent. Analog o Dig.
		TLC540	15 μs/can	8	11	—	Switc. Cap.	CMOS	20 pin DIP	como 532 pero salida serie
		TLC1540	20 μs/can	10	11	—	Switc. Cap.	CMOS		Salida serie
		TMS32050	62 μs	16	1	—	Aprox. Suc.	NMOS		Muestreo de la entrada
General	AMD	6080	0,16 μs	—	1	8	Red resist.	Bipolar	20 pin DIP	
		6081	0,2 μs	—	1	8	Red resist.	Bipolar	24 pin DIP	
		6012	0,25 μs	—	1	12	Red resist.	Bipolar	20 pin DIP	Resistencias difusas
		6112	7 μs	—	1	12	Red resist.		24 pin DIP	
General	Analog Devices	AD7226	5 μs/can	—	1	8	R-2R	LCCMOS	20 pin DIP	
		AD7522	1 μs	—	1	10	R-2R	CMOS	28 pin DIP	Carga serie o paralelo
		AD7524	0,1 μs	—	1	8	R-2R	CMOS	16 pin DIP	Mapeado en memoria
		AD7542/43	1 μs	—	1	12	R-2R	CMOS	16 pin DIP	Carga serie o paralelo
		AD7534	1,5 μs	—	1	14	R-2R	CMOS	20 pin DIP	Mapeado en memoria
12 bit		AD392	4 μs	—	4	12	Red resist.		32 pin	Híbrido, map. memoria
8 bit	Brooktree	BT110	0,1 μs/can	—	8	8	Curr. cels	CMOS	40 pin DIP	Pareado admite 12 bits
General 8-16 bit	Burr Brown	DAC705/7	8 μs	—	1	16	Red resist		24 pin DIP	
		DAC708/9	8 μs	—	1	16	Red resist		28 pin DIP	
		DAC74	50 μs	—	1	16	R-2R		40 pin	Híbrido, con autocalibración
4-8-12-16 bit		DAC811	3 μs	—	1	12	R-2R		28 pin DIP	
		DAC1201	7 μs	—	1	12	Red resist		28 pin DIP	
		PCM50	5 μs	—	1	16			24 pin DIP	
		AD7544		—	1	12				Con FIFO 6 niveles 12 bits
16 bit	DDC	DAC02310/1	1,8 μs	—	1	14	R-2R		32 pin	Híbrido
General	Ferranti	ZN558		—	1	8	R-2R			
CP1600	GI	DAC1600	0,3 μs	—	2	10	Anch. pulso	NMOS	40 pin DIP	
4-8-12-16 bit	Harris	HI5680/5/7	0,6 μs	—	1	12	R-2R	NMOS	24 pin DIP	
		HI5811	4 μs	—	1	12	R-2R	NMOS	28 pin DIP	

Tabla 7.4 (Continuación)

Compatibilidad bus-µC/bits	Fabricante	Referencia	Especificaciones: Tiempo de conversión	A/D bits	Canal	D/A bits	Técnica de conversión	Tecnología	n.º patillas (pin)	Comentarios
General	Intersil	ICL7145	3 µs	—	1	16	R-2R	NMOS	28 pin DIP	ROM interna de linealización
General	Micro Networks	MN3020	2,2 µs	—	1	8			118 pin	Híbrido
		MN3860	7 µs	—	1	12	Red resist.		24 pin DIP	
General 8 bit	PMI	PM8408	0,25 µs	—	1	8	R-2R	CMOS	28 pin DIP	
		DAC888	0,4 µs	—	1	8	R-2R	CMOS	18 pin DIP	
		PM7226	0,35 µs/can	—	4	8	R-2R	CMOS	20 pin DIP	
		PM7228	0,4 µs/can	—	2	8	R-2R	CMOS	20 pin DIP	
		PM7248		—	1	12	R-2R	CMOS	20 pin DIP	
4-8 bit		PM7242		—	1	12	R-2R	CMOS	16 pin DIP	
12 bit		PM7545/6	1 µs	—	1	12	R-2R	CMOS	20 pin DIP	
12-16 bit		DAC8012	1 µs	—	1	12	R-2R	CMOS	20 pin DIP	
		DAC8212	1 µs	—	1	12	R-2R	CMOS	24 pin DIP	
General	Signetics	NE/SE5410	<1 µs	—	1	10	R-2R	Bipolar	16 pin DIP	
		5028	2 µs	—	1	8	R-2R	Bipolar	22 pin DIP	

Tabla 7.4 (Continuación)

suelen tener unas prestaciones superiores en cuanto a calidad y precisión. La tabla 7.4 indica algunos de los convertidores actuales en el mercado y sus principales características.

7.6.1 Realización de conversiones por programa

En aplicaciones donde la velocidad de conversión no es muy importante y además, el microprocesador no está demasiado ocupado, éste puede tomar parte en la realización de conversiones controladas por programa.

Aunque hay varios caminos posibles para realizar la conversión A/D por software, la técnica de aproximaciones sucesivas parece la más adecuada en la mayoría de los casos, ya que ofrece una velocidad de conversión razonable utilizando un mínimo de circuitería adicional, consistente en un convertidor D/A y un comparador (figura 7.32). Es conveniente la utilización de un circuito de captura y mantenimiento que «congele» la señal de entrada mientras se realiza la conversión. El algoritmo de conversión puede ocupar entre 20 y 40 palabras de memoria, dependiendo, sobre todo, de la facilidad del microprocesador concreto, para testar bits. El comparador, compara la señal de entrada «congelada» con la salida del convertidor digital-analógico, y su resultado (0 si es menor, 1 si es mayor) retorna al procesador a través de un «port» o a través de un bit de prueba. El µP va enviando al convertidor D/A las palabras de aproximación, hasta que la conversión se ha realizado. La tabla 7.5 muestra una realización del algoritmo de aproximaciones sucesivas.

```
CAD:  MOV   R7, 8H     ; Resolución 8 bits
      CLR   A          ; Inicializacion
      MOV   R5,A       ; »
      MOV   R6, A      ; »
      CLR   C          ; »
      CPL   C          ; »
      ANL   P2, 0FEH   ; Mantenimiento S and H
BUC:  MOV   A, R5      ; bit de test
      RRC   A          ; Decalado
      MOV   R5, A
      ORL   A,R6       ; Almacena
      OUTL  P1, A
      JT0   TNV        ; Test nuevo valor
      MOV   R6, 4
TNV:  DJNZ  R7, BUC    ; busca nuevo bit
```

Tabla 7.5 Programa de conversión A/D y para un µC 8048

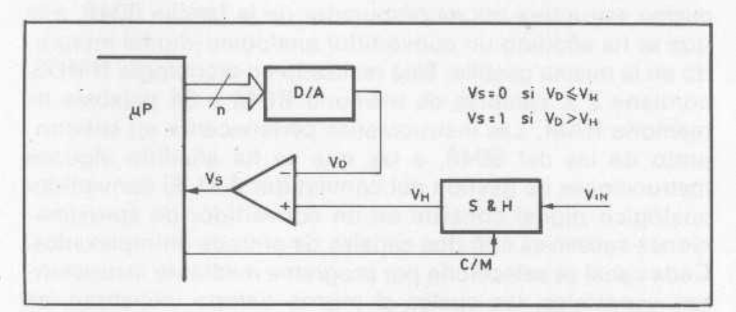


Figura 7.32 Circuito para realizar conversiones A/D mediante aproximaciones sucesivas por programa

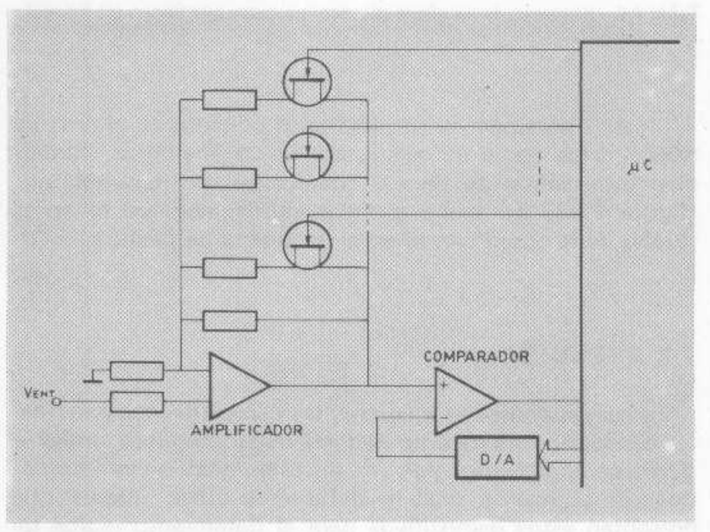


Figura 7.33 Conversión A/D con autoescalado

Fabricante	Referencia	µP/µC	Tecnología	Frecuencia MHz	Tiempo instr. min/máx	Long. pala. dat/inst.	ROM inter	RAM inter	Num. Instrucción	Can/bits A/D	Can/bits D/A	Patillas E/S	Num. Interrupciones	Stack	Aritmética BCD	Num. patillas	Alimentación
Fujitsu	MB88411	µC	NMOS	2	6/3	4/8	2K	192×4	75	sí	no	33	4	8	sí	42	5V
	MB88413	µC	NMOS	2	6/3	4/8	4K	192×4	75	sí	no	33	4	8	sí	42	5V
	MB88418	µP	NMOS	2	6/3	4/8	ext.	192×4	75	8/8b	no	—	sí	RAM	sí	42	5V
	MB88419	µP	NMOS	2	6/3	4/8	ext.	192×4	75	8/8b	no	—	sí	RAM	sí	82	5V
GTE Micr.	HD63L05F	µC	CMOS	4	4/2	8/8	3772×8	96×8	61	sí	no	20	2	RAM	sí	60	3V
Hitachi	HD6805W	µC	NMOS	4	4/2	8/8	3848×8	96×8	61	sí	no	29	2	RAM	sí	40	5V
Intel	8022	µC	NMOS	3,58	16,8/ 8,4	8/8	2K	64×8	70	2/8	no	27	1	8	sí	40	5V
	8096	µC	NMOS	12	6,5/1	16/16	4K×1	16×16	95	8/10	no	24	8	RAM	no	48	5V
	2920	µC	NMOS	6,67	0,6/0,6	25/24	192×24	40×25	21	4/9	8/8	12	no	no	—	28	±5V
	2921	µC	NMOS	10	0,4/0,4	25/24	192×24	40×25	21	4/9	8/8	12	no	no	—	28	±5V
Mitsubishi	M50734	µP	CMOS				Externas			4/8b	1/5b	32					
Electron.	M50744	µC	CMOS			8b	4Kb	144b		4/8b	1/5b	53					
	M50746/ES	µC	CMOS			8b	6Kb	144b		4/8b	1/5b	53					
Motorola	MC6805R2	µC	HMOS	5	4/2	8/8	2Kb	64×8	59	4/8	no	32	1	RAM	sí	40	5V
	MC6805R3	µC	HMOS	5	4/2	8/8	3776×8	112×8	59	4/8	no	32	1	RAM	sí	40	5V
	MC68705R3	µC	HMOS	5	4/2	8/8	3776×8	112×8	59	4/8	no	31	1	RAM	sí	40	5V
National	HPC16205	µP	CMOS	20		16	ext.	1Kb		sí	no		8	RAM	sí		
Semicond.	HPC16140	µC	CMOS	20		16	4Kb	256b		sí	no		8	RAM	sí		
	HPC16E180	µC	CMOS	20		16	8Kb	256b		sí	no		8	RAM	sí		
	COP444L	µC	NMOS	0,26	40/15	4/8	2K	128×4	49	sí	no	23/35	1	3	sí	28/40	4,5–9V
	COP445L	µC	NMOS	0,26	40/15	4/8	2K	128×4	49	sí	no	19	1	3	sí	24	4,3–6,3V
NEC Elec.	µPD7811	µC	NMOS	1	4/2	8/8	4K	128×8	160	8/8	no	48	5	RAM	sí	64Q	5V
SGS	M38AS72	µC	NMOS	4			4Kb	128b		4/8b		32	4			48/68	5V
Siemens	SAB80515/35	µC				8/8	8K	256×4		8/8	—	48	sí	sí		68Q	5V
Texas	TMS2100	µC	PMOS	0,55	60/11	4/8	2K	128×4	55	1/8	no	20	1	4	sí	28	–9V
	TMS2170	µC	PMOS	0,55	60/11	4/8	2K	128×4	55	1/8	no	19	1	4	sí	28	–9V
	TMS2300	µC	PMOS	0,55	60/11	4/8	2K	128×4	55	2/8	no	33	1	4	sí	40	–9V
	TMS2370	µC	PMOS	0,55	60/11	4/8	2K	128×4	55	2/8	no	32	1	4	sí	40	–9V
	TMS2400	µC	PMOS	0,55	60/11	4/8	4K	256×4	55	1/8	no	20	1	4	sí	28	–9V
	TMS2470	µC	PMOS	0,55	60/11	4/8	4K	256×4	55	1/8	no	19	1	4	sí	28	–9V
	TMS2600	µC	PMOS	0,55	60/11	4/8	4K	256×4	55	4/8	no	33	1	4	sí	40	–9V
	TMS2670	µC	PMOS	0,55	60/11	4/8	4K	256×4	55	4/8	no	32	1	4	sí	40	–9V
Toshiba	IMP8022	µC	NMOS	3,6	16,8/ 8,4	8/8	1K	64×8	70	sí	no	13	no	8	sí	20	5V

Tabla 7.6 Resumen de los chips microcomputador que tienen incorporados convertidores analógico-digitales o digital-analógicos en su estructura interna

La incorporación de posibilidades adicionales al circuito básico, tales como autoescalado, salida analógica, corrección automática de derivas, detección de max-min, etc. (figura 7.33) no exige grandes modificaciones ni en el diseño de la circuitería ni en el programa de gestión.

7.7 APENDICE I

En este apéndice se muestran las características principales de algunos microcomputadores que incorporan convertidores analógico-digitales y/o digitales-analógicos integrados en la misma pastilla debido al obvio interés que presentan estos dispositivos para el tratamiento digital de señales analógicas (tabla 7.6).

7.7.1 El µC 8022

La arquitectura de este componente está basada en la misma estructura microcomputador de la familia 8048, a la que se ha añadido un convertidor analógico-digital integrado en la misma pastilla. Está realizado en tecnología NMOS, contiene 2 K palabras de memoria ROM y 64 palabras de memoria RAM. Las instrucciones pertenecen a un subconjunto de las del 8048, a las que se ha añadido algunas instrucciones de gestión del convertidor A/D. El convertidor analógico-digital consiste en un convertidor de aproximaciones sucesivas con dos canales de entrada multiplexados. Cada canal se selecciona por programa mediante instrucciones especiales, las cuales al mismo tiempo inicializan las secuencias de conversión. Una vez seleccionado el canal, se realizan conversiones de una manera continua, las cuales

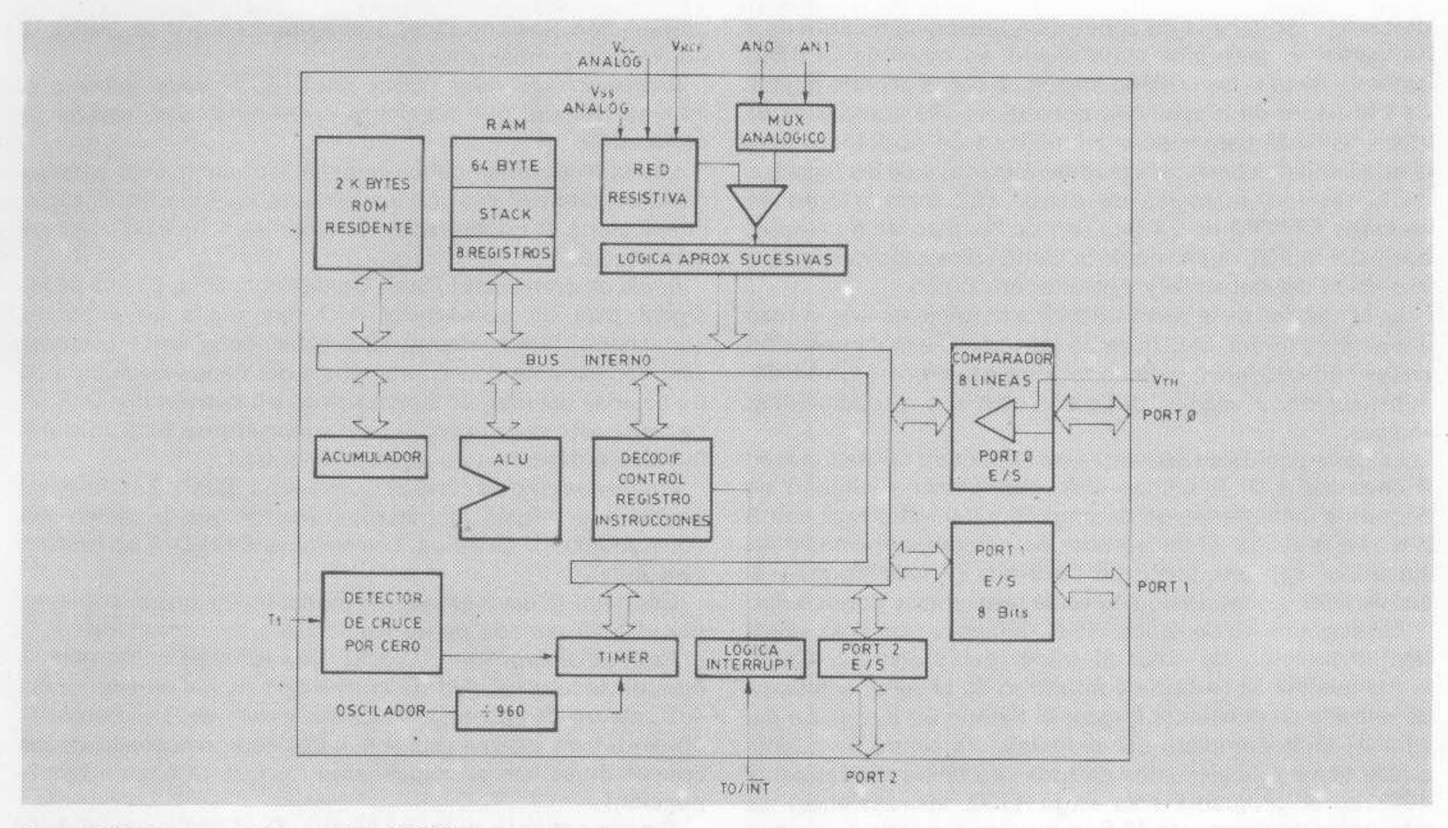


Figura 7.34 Esquema de bloques del μC 8022

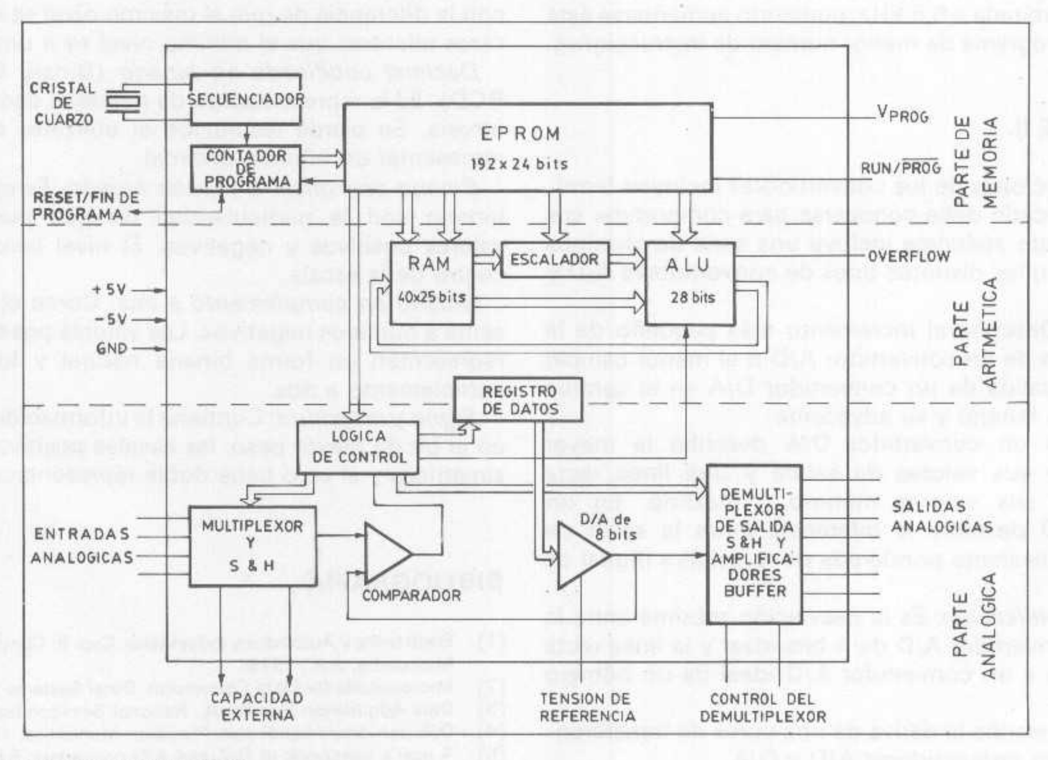


Figura 7.35 Diagrama funcional del μC 2920

pueden ser leídas con la seguridad de que la medida no es anterior a 40 microsegundos, que es el tiempo de conversión. Por programa se pueden seleccionar hasta 27 señales de entrada o salida incorporadas en la misma pastilla. Tiene un único nivel de interrupción. La figura 7.34 muestra un esquema de bloques de este dispositivo.

7.7.2 El μC 2920

Este microcomputador presenta una estructura totalmente dirigida al procesado de señales analógicas. Su arquitectura puede dividirse en tres partes principales (figura 7.35). La parte de memoria de programa, la parte aritmética o de

procesado digital y la de conversión analógica de entrada y de salida. El programa almacenado en memoria EPROM controla tanto la conversión analógica como la parte digital. La estructura de la memoria consiste en 40 posiciones de memoria RAM para realizar aritmética y 24 posiciones para la selección de constantes predeterminadas y de los registros de la sección analógica de datos; 192 instrucciones de memoria EPROM de 24 bits con un formato de 5 campos; operador digital, dirección de la fuente, dirección del destino extensión del decalado y operador analógico.

La estructura de la parte aritmética consiste en una unidad aritmética y lógica (ALU) de 25 bits, con 9 bits de precisión en los convertidores, permite movimiento de datos, adición, substracción y cálculo del valor absoluto y operaciones lógicas.

La parte analógica presenta una estructura consistente en 4 entradas A/D, 8 salidas D/A, multiplexor y circuito de captura y mantenimiento de entrada y demultiplexor con 8 S & H a la salida. El convertidor A/D es de aproximaciones sucesivas, con una precisión de 9 bits (8 más signo) y el convertidor digital-analógico es de resistencias ponderadas.

El programa es de único ciclo, debido a que sólo existe una instrucción de salto al inicio del programa. Como consecuencia, el tiempo de muestreo de la señal analógica de entrada es constante e igual al tiempo de ejecución del (único) ciclo completo del programa. Teniendo en cuenta que el tiempo de ejecución de todas las instrucciones es de 400 ns, el programa más largo (192 instrucciones) se ejecuta en un tiempo de 76,8 μs que corresponde a un ritmo de muestreo aproximado de 13 kHz. De acuerdo con el teorema del muestreo, la anchura de banda de la señal de entrada queda limitada a 6,5 kHz, pudiendo aumentarse ésta realizando un programa de menor número de instrucciones.

7.8 APENDICE II

Las especificaciones de los convertidores incluyen términos cuyo significado debe conocerse para comprender sus posibilidades. Este apéndice incluye una serie de términos de uso común en los distintos tipos de convertidores A/D y D/A.

Resolución: Describe al incremento más pequeño de la señal de entrada de un convertidor A/D o el menor cambio de la señal de salida de un convertidor D/A en el cambio entre un código binario y su adyacente.

Precisión: En un convertidor D/A describe la mayor diferencia entre sus valores de salida y una línea recta dibujada entre sus valores mínimo y máximo. En un convertidor A/D describe la diferencia entre la señal de entrada y la equivalente ponderada de la palabra digital de salida.

Error de cuantificación: Es la desviación máxima entre la salida de un convertidor A/D de *n* bits ideal y la línea recta correspondiente a un convertidor A/D ideal de un número infinito de bits.

Linealidad: Describe la deriva de una curva de transferencia lineal para los convertidores A/D y D/A.

Monotonicidad: Describe la falta de continuidad entre la característica de transferencia de un convertidor D/A y los incrementos de la salida, que deberían ser constantes, al incrementarse la entrada.

Tiempo de conversión: Es el tiempo que necesita un convertidor para la realización de una conversión cuando realiza conversiones consecutivas. Incluye el tiempo de preparación (settling time) y el tiempo en que se realiza la conversión propiamente dicha.

Frecuencia de reloj (clock rate): Es el valor máximo o mínimo de reloj que admite el convertidor para realizar la conversión.

Los convertidores deben aceptar y/o suministrar señales digitales, compatibles con los sistemas externos con los que interacciona. A continuación se citan las señales de control más usuales.

Inicio de conversión (Start Conversion, SC): Es una señal digital para un convertidor A/D que inicia un ciclo de conversión. Normalmente esta señal debe estar presente simultáneamente con la ocurrencia del flanco de subida (o de bajada) del reloj de sincronismo. Un convertidor D/A no necesita esta señal, pero puede suministrarse para capturar la palabra digital en su registro de entrada.

Fin de conversión (End of Conversion, EOC): Es una señal digital que infoma al microprocesador que la salida del convertidor A/D es válida. Los convertidores D/A no utilizan esta señal.

Ocupado (Busy): Esta señal digital indica que el convertidor está en período de realización de una conversión.

Reloj de sincronismo (Clock): Esta señal se utiliza para el control de la velocidad de conversión de los convertidores A/D, dentro de sus límites máximo y mínimo. Los convertidores D/A no utilizan esta señal. Los códigos usuales de los convertidores son principalmente los que se citan a continuación:

Binario natural o unipolar binario: Es el código usual de la progresión 2^n.

Binario complementario: Es la misma progresión anterior, con la diferencia de que el máximo nivel se representa por *n* ceros mientras que el mínimo nivel es *n* unos.

Decimal codificado en binario (Binary Coded Decimal, BCD): Es la representación de números decimales en forma binaria. Se pierde resolución al utilizarse cuatro bits para representar un número decimal.

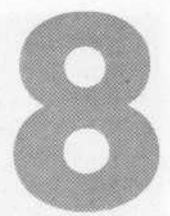
Binario con offset o bipolar binario: Es el código natural binario con la particularidad de que puede representar valores positivos y negativos. El nivel cero se sitúa en el centro de la escala.

Binario en complemento a dos: Como el anterior, representa a números negativos. Los valores positivos y el cero se representan en forma binaria natural y los negativos en complemento a dos.

Signo y magnitud: Contiene la información de la polaridad en el bit de mayor peso; las escalas positiva y negativa son simétricas y el cero tiene doble representación.

BIBLIOGRAFIA

[1] Electrónica y Automática industriales. Cap. 8: Convertidores A/D y D/A. Marcombo, S.A., 1979.
[2] Microcircuits for Data Conversion. Datel Systems Inc. 1978.
[3] Data Adquisition Handbook. National Semiconductor. 1977.
[4] D/A converters handbook. Precision Monolitics. 1977.
[5] A user's handbook of D/A and A/D converters. E.R. Hnatek. Ed. John Wiley. 1976.
[6] Digital and Analog Data Conversion. Malmstadt. Ed. W.A. Benjamin. 1973.
[7] 8022 Data Sheet; 2920 Data Sheet; INTEL 1978/79.
[8] Linear electronic analog-digital conversion architectures, Gordon B.M. IEEE Transactions on Circuits and Systems Vol. CAS-25 n.º 7 Julio 1978.
[9] Analog-Digital Conversion Handbook. Analog Devices. Prentice-Hall, Englewood Cliffs, N.J. 1986.



teclados

Joan Navarro Blanco

8.1 GENERALIDADES

Se denomina «teclado» al género de periféricos de entrada, constituidos por un conjunto de botones pulsadores, de tal modo que cada botón se corresponda con determinado carácter, función, instrucción o idea.

El tipo de teclas, así como su número y distribución, vendrán determinados por la aplicación concreta que se desee realizar, por lo que no existen modelos genéricos, sino desarrollos específicos.

El número de teclados conectados a un determinado sistema es sumamente variable, oscilando entre cero (tal como en sistemas muy simples de instrumentación o automatización) y varias decenas (tal como en sistemas multiterminal de recogida y consulta de bancos de datos); no obstante, en sistemas basados en microprocesadores, lo más común es disponer de una unidad, desde donde se suministran al sistema las informaciones básicas en cuanto a selección y control de programas, e introducción de variables.

Físicamente, el teclado acostumbra a ir asociado a otro periférico de salida, tal como una impresora, una pantalla o un visualizador, con lo que el operador obtiene una comunicación bidireccional con el sistema.

8.2 PULSADORES

El componente básico de un teclado es el pulsador individual.

Generalmente, cada tecla controla un simple interruptor que permanece abierto mientras el pulsador está en descanso y se cierra cuando el pulsador es deprimido.

Existe un sinnúmero de variaciones, relacionándose más adelante los tipos más usuales.

El concepto de conmutación de estado sólido se aplica a veces al describir elementos sin contacto, aunque hablando con propiedad solamente se puede aplicar a una limitada variedad entre la que se cuentan los de efecto Hall o elementos fotosensibles.

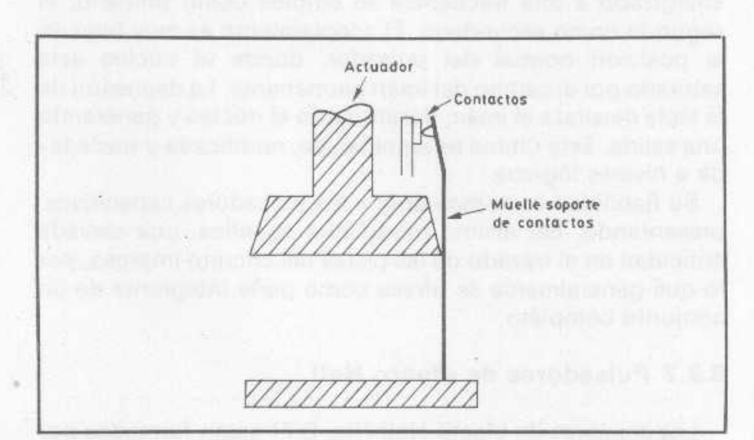


Figura 8.1 Pulsador mecánico

8.2.1 Pulsadores mecánicos convencionales

Los interruptores mecánicos son utilizados básicamente por su economía (fig. 8.1).

El movimiento mecánico actúa directamente sobre los contactos.

Permiten, sin grandes complicaciones, configuraciones de contactos múltiples.

8.2.2 Pulsadores mecánicos de lámina flexible

Este tipo de interruptor (fig. 8.2), formado por una serie de láminas sobrepuestas, se basa en la deflexión de un diafragma flexible, dorado por su cara inferior, que permite establecer contacto con un circuito impreso a través de aperturas practicadas en un separador dieléctrico.

Una cubierta de silicona protege los contactos contra los contaminantes. Algunas versiones más económicas emplean láminas flexibles de silicona conductora, que sustituyen la cubierta protectora y el diafragma dorado. En algunos casos se emplea una base serigrafiada de tinta conductora como sustrato.

8.2.3 Pulsadores mecánicos de bovedillas

Estos pulsadores (fig. 8.3) emplean unos discos metálicos embutidos en forma de casquete esférico, que cuando son deprimidos pasan a una condición invertida, estableciendo el contacto requerido.

Emiten un clásico chasquido audible que advierte de su correcta operación.

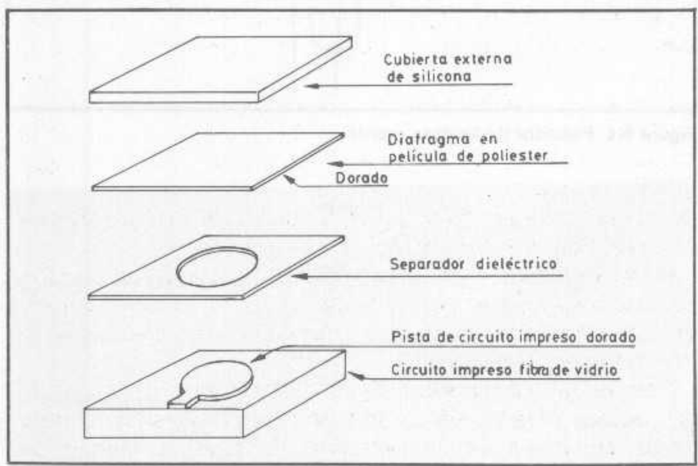


Figura 8.2 Pulsador de lámina flexible

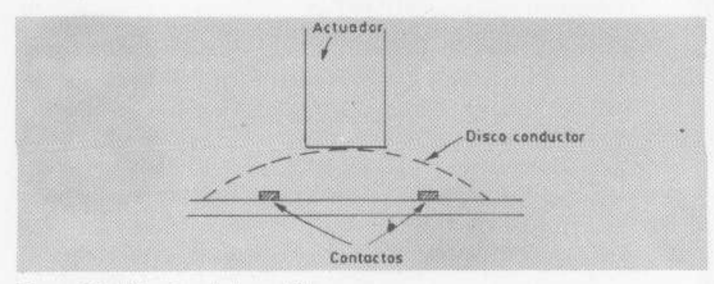


Figura 8.3 Pulsador de bovedilla

8.2.4 Pulsadores reed

Están formados por contactos inmersos en una atmósfera inerte, sellados por una cápsula de vidrio hermética (fig. 8.4).

El movimiento de la tecla desplaza un pequeño imán permanente que provoca el cierre de los contactos; dada la acción indirecta sobre estos últimos, no se transmiten sobrecargas mecánicas que provoquen fatiga y desgastes prematuros. Por lo anterior y por la hermeticidad del encapsulado que impide la contaminación de los contactos, este tipo de pulsadores ofrecen una vida útil unas 5 veces superior al clásico pulsador mecánico.

8.2.5 Pulsadores capacitivos

Este tipo de pulsadores emplea un cambio en la capacidad de un condensador para entregar una salida (fig. 8.5).

Emplean dos superficies vecinas sobre un mismo circuito impreso, estando una de ellas excitada por la señal alterna de un oscilador; si se aproxima paralelamente una placa conductora sobre ambas superficies, se provoca un acoplamiento entre ellas, con lo que aparece una fracción de la señal alterna en la salida.

La señal de salida del pulsador debe ser convenientemente amplificada y convertida a niveles lógicos.

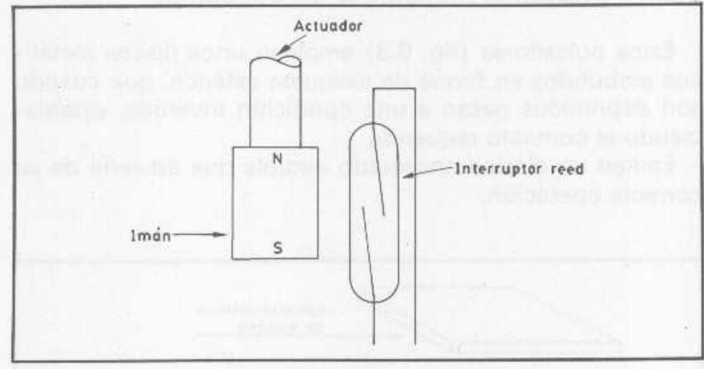


Figura 8.4 Pulsador de láminas (reed)

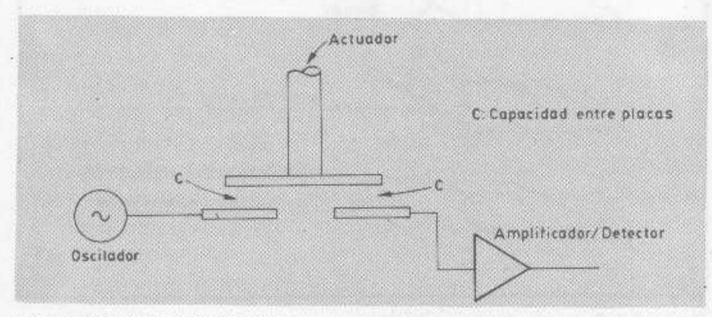


Figura 8.5 Pulsador capacitivo

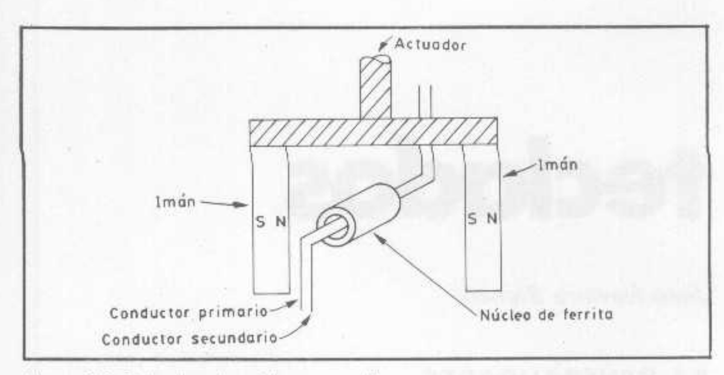


Figura 8.6 Pulsador de núcleo magnético

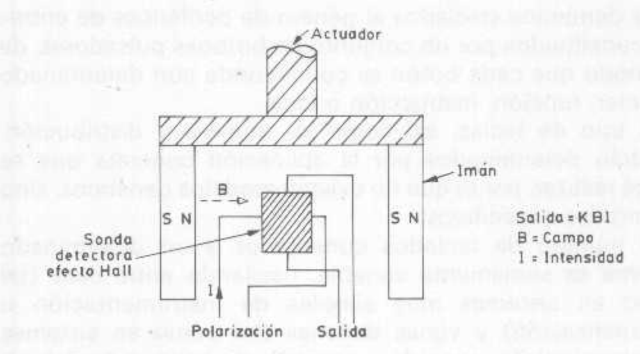


Figura 8.7 Pulsador de efecto Hall

Existen múltiples variantes tales como los que utilizan contactos sensitivos, sin ningún elemento móvil. Otros diseños emplean bovedillas metálicas cóncavas como elemento de acoplamiento.

Los pulsadores capacitivos ofrecen la elevada fiabilidad de los interruptores sin contactos móviles.

Dados los bajos niveles de señal entregada por estos pulsadores se presenta una acusada sensibilidad a interferencias y unos serios condicionamientos en la estructura metálica soporte, trazado de pistas en el circuito impreso y electrónica de amplificación, detección y conversión. Por todo ello sólo aparecen disponibles formando parte de teclados completos producidos por fabricantes especializados.

8.2.6 Pulsadores de núcleos magnéticos

El elemento conmutador es un núcleo toroidal de ferrita empleado como transformador (fig. 8.6).

El núcleo es atravesado normalmente por 2 hilos: uno, energizado a alta frecuencia se emplea como primario; el segundo como secundario. El acoplamiento es muy bajo en la posición normal del pulsador, donde el núcleo está saturado por el campo del imán permanente. La depresión de la tecla desplaza el imán, desaturando el núcleo y generando una salida. Esta última es amplificada, rectificada y trasladada a niveles lógicos.

Su fiabilidad es comparable a los pulsadores capacitivos, presentando, del mismo modo que aquellos, una elevada criticidad en el trazado de las pistas del circuito impreso, por lo qué generalmente se ofrece como parte integrante de un conjunto completo.

8.2.7 Pulsadores de efecto Hall

Los sensores de efecto Hall (fig. 8.7) están formados por una pastilla semiconductora recorrida por una corriente

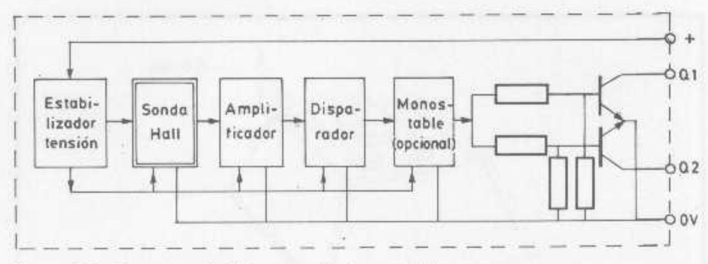


Figura 8.8 Esquema de bloques del detector Hall

continua, y un campo magnético perpendicular a ella que provoca una deformación de las líneas equipotenciales sobre la superficie del semiconductor, apareciendo una tensión de salida proporcional al producto de la corriente de polarización por la intensidad del campo magnético aplicado.

La conmutación se obtiene al aproximar un imán permanente al sensor, que desarrolla una tensión de salida que es amplificada y convertida en digital.

Generalmente el conjunto formado por el sensor, amplificador, disparador de Schmitt, monostable opcional y paso de salida, forma un circuito integrado monolítico (fig. 8.8) asociado a cada pulsador.

Se distinguen dos tipos fundamentales: estático y dinámico. Los pulsadores estáticos conducen a su salida, mientras exista campo magnético a su entrada. Los pulsadores dinámicos conducen durante cierto período (típicamente 20 μs) cuando el campo de entrada supera el nivel de conmutación, pero no lo hacen durante el resto del tiempo que dicho campo permanezca a nivel elevado, ni durante el alejamiento del imán; para ello incorporan un monostable que dispara en el flanco de subida de la señal magnética.

La salida puede estar formada por un transistor de colector abierto, simple o doble, o bien por una puerta lógica «Y» aceptando señales externas de validación y sincronismo.

Dada la ausencia de contactos, la baja impedancia de todas las señales de interconexión y la insensibilidad a polvo, suciedad y contaminantes, este tipo de pulsadores ofrece la mayor fiabilidad (esencialmente duración infinita), sólo limitada por el desgaste del elemento móvil y el resorte de retorno. Este último, en algunos casos, es sustituido por un sistema magnético de retorno, que proporciona simultáneamente una realimentación al tacto.

8.2.8 Pulsadores inductivos

Los pulsadores inductivos parten de un principio de funcionamiento situado a caballo de los equivalentes de los capacitivos y de los de núcleos magnéticos.

Este tipo de pulsadores se basa en una variación de la permeabilidad magnética en el medio de acoplamiento de dos circuitos inductivos (fig. 8.9).

Los dos circuitos inductivos están definidos a ambas caras de un mismo circuito impreso. Uno de dichos circuitos está recorrido por una corriente de alta frecuencia (impulsos), mientras que el otro podrá captar dicha señal si existe un buen acoplamiento.

El acoplamiento a través del aire y del propio substrato del circuito impreso es muy débil, por lo que el nivel de señal obtenido a la salida de los secundarios asociados a las teclas no pulsadas es suficientemente bajo como para ser discriminado como «0» lógico. La depresión de la tecla introduce un núcleo de alta permeabilidad magnética (ferrita) en el orificio del circuito impreso situado en el eje común a ambas espiras, con lo que el acoplamiento se incrementa de forma

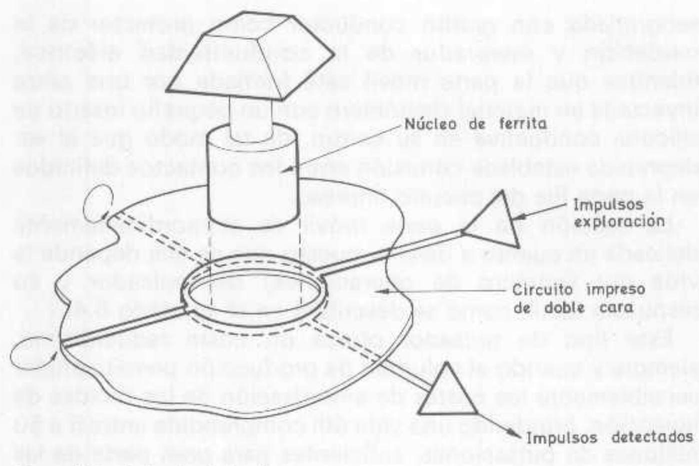


Figura 8.9 Pulsador inductivo

notable, induciendo una corriente secundaria (como consecuencia de cada impulso que recorre el circuito primario) de suficiente nivel como para ser discriminada como de nivel lógico «1».

Su fiabilidad es tan elevada como la del resto de técnicas que no implican un contacto físico, es decir: capacitivos, núcleos magnéticos o de efecto Hall. Actualmente se están imponiendo de forma notable dada su combinación de alta fiabilidad y coste reducido, siendo este último consecuencia de no precisar un circuito integrado para cada tecla (como en el caso del efecto Hall), ni un cableado con alta incidencia del coste de mano de obra (como en el caso de los pulsadores de núcleos magnéticos), ni presentar la criticidad de diseño asociada a los pulsadores capacitivos (téngase presente que el circuito correspondiente a cada secundario, usualmente un grupo de 10 a 16 teclas, es un circuito cerrado de muy baja impedancia, por lo que la sensibilidad a interferencias de origen externo está sensiblemente reducida).

8.2.9 Pulsadores elastómeros

Estos pulsadores no constituyen estrictamente un grupo nuevo con tecnología propia, sino una variante tecnológicamente evolucionada de los pulsadores de lámina flexible (fig. 8.10).

Están constituidos por un circuito impreso, en el que están definidos tantos pares de contactos como debe contener el conjunto del teclado, y un elemento elastómero (silicona) que forma la parte móvil del contacto.

La superficie de contacto del circuito impreso está

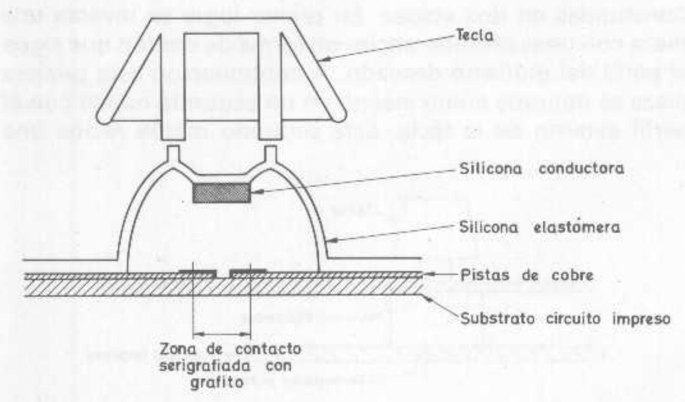


Figura 8.10 Pulsador elastómero

serigrafiada con grafito conductor como protector de la oxidación y mejorador de la conductividad eléctrica, mientras que la parte móvil está formada por una pieza inyectada en material elastómero con un pequeño inserto de silicona conductiva en su centro, de tal modo que al ser deprimida establece conexión entre los contactos definidos en la parte fija del circuito impreso.

La sección de la parte móvil es extraordinariamente delicada en cuanto a diseño, puesto que de ella depende la vida útil (número de operaciones) del pulsador y su respuesta táctil, como se describirá en el apartado 8.4.

Este tipo de pulsador ofrece un coste reducidísimo, siempre y cuando el volumen de producción permita anular sensiblemente los costes de amortización de los moldes de inyección, brindando una vida útil comprendida entre 5 a 50 millones de pulsaciones, suficientes para gran parte de las aplicaciones.

8.3 TECLAS

El elemento unitario componente de un teclado, es el formado por el subconjunto: pulsador más tecla, que a menudo es denominado simplemente tecla (fig. 8.11).

La tecla o capuchón, aun cuando no cumple ninguna función eléctrica, es absolutamente imprescindible con el fin de identificar adecuadamente cada pulsador, así como ofrecer una correcta superficie de actuación apropiada a las características fisiológicas de manos y dedos del operador.

8.3.1 Rotulación

Con el fin de obtener una correcta identificación de cada uno de los distintos pulsadores que componen un teclado es preciso rotular con los grafismos apropiados cada una de las teclas.

El grafismo debe facilitar una rápida localización de la tecla idónea, por lo que debe ser claro y esquemático. Los grafismos correspondientes a las teclas asociadas a caracteres alfanuméricos no presentan evidentemente ningún problema; en cambio, las diversas funciones de control han exigido el desarrollo de una serie de grafismos y abreviaturas un tanto criptográficas para el no iniciado.

Para fijar sobre la tecla su correspondiente grafismo se utilizan distintas técnicas, siendo las más usuales: serigrafía, pantografiado, doble inyección y empleo de teclas transparentes rotulables.

Los métodos serigráfico y pantográfico son similares a los empleados para marcado de paneles metálicos, circuitos impresos, etc., por lo que no procede mayor ampliación.

Las teclas rotuladas por doble inyección (fig. 8.12) están construidas en dos etapas. En primer lugar se inyecta una pieza con unas protuberancias en forma de cordón que sigue el perfil del grafismo deseado. A continuación esta primera pieza es utilizada como inserto en un segundo molde con el perfil externo de la tecla. Este segundo molde recibe una

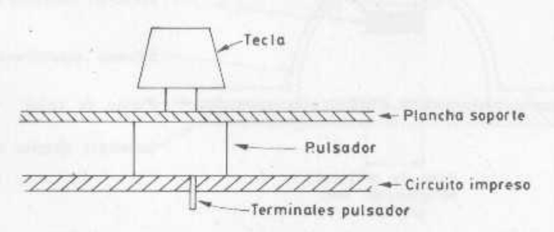


Figura 8.11 Conjunto pulsador más tecla

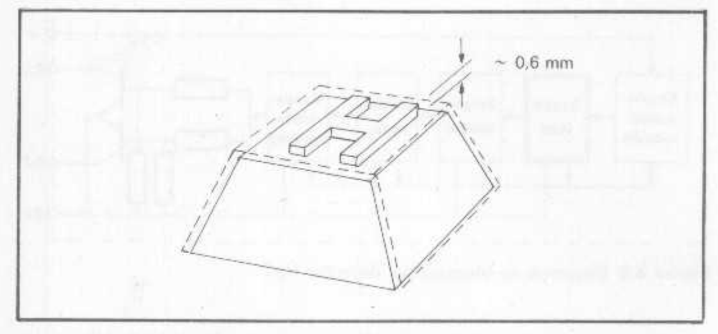


Figura 8.12 Primera fase de una tecla de doble inyección

inyección de plástico con distinto colorante del inserto, por lo que éste aparece finalmente como un afloramiento de la cresta de la primera inyección, definiendo por tanto, por medio de un color distintivo, el rótulo deseado.

Las rotulaciones serigrafiadas y pantografiadas adolecen del defecto de ser sumamente sensibles a la erosión causada por el roce mecánico con los dedos del operador y de los agentes químicos, entre los que deben contarse como más importantes el sudor humano y los líquidos limpiadores. Esta sensibilidad queda totalmente compensada mediante el empleo de doble inyección, por cuanto el plástico utilizado es insoluble a los agentes químicos ambientales y la eventual erosión mecánica provoca un desgaste paralelo tanto en la primera inyección (grafismo) como en la segunda (cuerpo) por lo que mientras esta erosión no desgaste más allá de medio milímetro de plástico, la tecla aparecerá como perfectamente legible.

Las ventajas mencionadas de las teclas doblemente inyectadas tienen su contrapartida bajo el punto de vista económico, de tal modo que el costo de los utillajes sólo puede ser amortizado bajo unas condiciones de cantidad y exigencias de calidad, que no siempre se cumplen; para solventar estas situaciones (fundamentalmente las series reducidas o prototipos) se han desarrollado unas teclas especiales con la cubierta total o parcialmente transparente. Esta ventana (o la cubierta completa, según sea el caso) es desmontable, y permite la inserción de leyendas rotuladas o impresas sobre papel en forma de etiquetas, facilitando de forma excepcional la obtención de teclados especiales en un tiempo mínimo.

Recientemente se ha desarrollado una interesante alternativa a la técnica de doble inyección denominada «sublimación». El método de sublimación consiste en transferir mediante presión y temperatura las leyendas, residentes en una hoja de transferencia en la que han sido impresas por medios fotosensibles, a la superficie superior de las teclas donde penetren algunas micras. Como ventajas del método se pueden considerar su bajo coste (la grabación de todas las teclas del conjunto de un teclado se efectua en una sola operación), su flexibilidad (que permite obtener nuevos grafismos o modificar los presentes sin la elevada inversión en tiempos y dinero que supone la doble inyección) y su elevada resistencia tanto química como mecánica. Entre sus defectos cabe considerar el hecho de que los capuchones de las teclas deben ser de poliéster (a fin de permitir la penetración del tinte, lo que no se obtiene en ABS u otros plásticos industriales de mejor resistencia), que el color de la tinta de los grafismos debe ser más obscuro que el del cuerpo de las teclas (puesto que se transparenta a través de la película superficial), y que plantea serias problemáticas para obtener la impresión en varios colores. No obstante, el

balance es suficientemente favorable como para que los principales fabricantes se hayan decantado mayoritariamente por la sublimación.

Como complemento a los sistemas de rotulación de teclas deben considerarse otros sistemas de identificación, como el empleo de plástico de distintos colores para diferenciar teclas de especial interés; teclas luminosas mediante diodos LED mono o bicolor o bien mediante lamparitas que iluminan por transparencia rótulos translúcidos; distintas dimensiones de la usual cuadrada de 19 mm (3/4''), que se denomina usualmente de una unidad, existiendo comúnmente teclas de 1, 1 1/4, 1 1/2, 1 3/4, 2, 3, 6, 8 y 9 unidades de longitud, así como figuras distintas del cuadrilátero, siendo relativamente normales formas de «L».

8.3.2 Perfil

El conjunto formado por la superficie superior de cada una de las teclas puede adoptar distintas configuraciones basadas en las preferencias y aplicaciones deseadas.

Los teclados suelen montarse formando un ángulo de 10° a 13° con la horizontal.

La interacción entre orientación de teclas e inclinación general del teclado dará lugar a las configuraciones antes indicadas, y que pueden resumirse en tres tipos fundamentales (fig. 8.13).

—Inclinado («Sloped»).

—Escalonado («Stepped»).

—Esculpido («Sculptured»)

El teclado inclinado (fig. 8.13*a*) está formado por un conjunto de pulsadores y teclas totalmente verticales en origen, que posteriormente se giran, adoptando la inclinación general del teclado, con lo que las superficies superiores de las teclas son coplanarias (salvando la ligera concavidad asociada a cada tecla).

El teclado escalonado (fig. 8.13*b*) está formado por un conjunto de pulsadores y teclas no alineados originalmente, sino formando un ángulo entre 10° a 12° (sea por cambio de orientación del vástago del pulsador, sea por modificación del perfil de la tecla). Al girar el conjunto, dándole la inclinación general, las superficies superiores de las teclas quedan horizontales, formando filas a distintas alturas, o escalones.

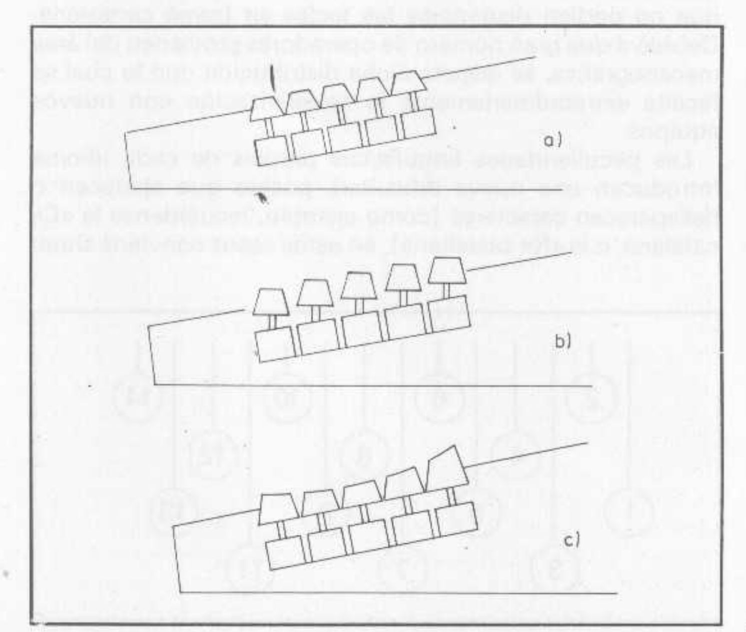


Figura 8.13 Perfiles de teclados: *a*) inclinado; *b*) escalonado; *c*) esculpido

El teclado esculpido (fig. 8.13*c*) presenta una cierta similitud con el escalonado, con la diferencia de que los ángulos iniciales de pulsadores y teclas no son comunes a la totalidad de elementos, sino función de la fila donde están situados, de tal modo que las filas superiores forman ángulos más acentuados que el general del teclado, mientras que las inferiores no sólo lo desacentúan, sino que llegan a adoptar pendiente negativa. El conjunto adopta con todo ello un perfil curvado.

Los teclados inclinados y escalonados, según diversos estudios, no aportan sustanciales beneficios en cuanto a velocidad, seguridad o comodidad del operador, por lo que se adoptan indistintamente; la costumbre asocia los teclados inclinados a las pequeñas calculadoras portátiles y los escalonados a las máquinas de escribir, por lo que según la aplicación tienda más hacia uno u otro sentido, se adopta la solución más próxima.

El teclado esculpido está justificado por razones ergonómicas, de tal modo que las distancias entre las yemas de los dedos y las superficies de apoyo de las teclas se uniformizan, presentando asimismo cada fila de teclas el ángulo de ataque más apropiado. Con todo ello se mejoran velocidad, fiabilidad y confort del operador, dando por tanto mejores rendimientos en aplicaciones de trabajo pesado. Se popularizó inicialmente a partir de los equipos perforadores de tarjetas, y se ha extendido su uso en todos los ámbitos de la informática de alto nivel.

La superficie superior de cada una de las teclas puede adoptar diversas configuraciones: plana, esférica o cilíndrica (fig. 8.14). La superficie plana se utiliza en aplicaciones sencillas, de bajo coste, sin requisitos estrictos en cuanto a ergonomía. La superficie esférica ha sido utilizada tradicionalmente en los teclados profesionales, dando lugar, mediante su concavidad, a una referencia de situación para los dedos del operador; esta referencia ha sido aumentada en algunas aplicaciones mediante la adición de una pequeña protuberancia en el centro de la concavidad de algunos grafismos, tales como el «5», la «F» y la «J». La superficie cilíndrica ha sido popularizada, como alternativa a la esférica, por IBM en su PC; conserva las virtudes de la superficie esférica y aporta una nueva estética que se está expandiendo fuertemente.

8.4 REALIMENTACION FISIOLOGICA

Es la variación de la fuerza de apoyo sobre la tecla, en función de su desplazamiento, que simula la conmutación de un interruptor (fig. 8.15).

En ocasiones se desea que el teclado simule totalmente una máquina de escribir, una perforadora de tarjetas, u otros equipos similares en que mecánicamente aparece el fenómeno de la respuesta táctil.

A menudo se desea asimismo un «clik» audible, que en muchas ocasiones es utilizado para suplantar la respuesta al



Figura 8.14 Superficie superior de las teclas

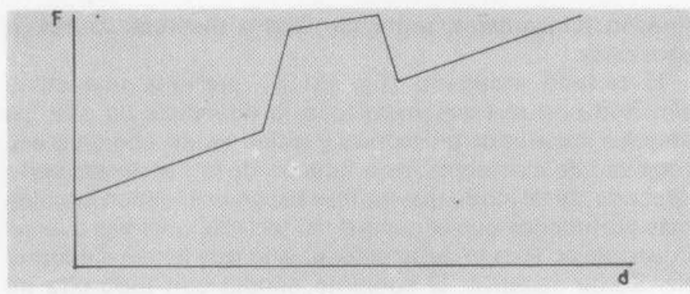


Figura 8.15 Realimentación táctil

tacto, sobre todo cuando se desea realimentación fisiológica en teclados sin contactos, que no lo generan por sí mismos.

Dado que los operadores se adaptan fácilmente a la ausencia del tacto del interruptor, no existe generalmente justificación técnica al incremento de costo y complejidad que supone la generación de la respuesta aural.

8.5 FORMATOS

El número de teclas y su distribución relativa es esencialmente variable y sólo puede concretarse para cada aplicación particular; no obstante existen en los catálogos de los fabricantes algunos modelos estándar que se adaptan a un elevado porcentaje de aplicaciones.

Uno de los tipos más comunes es el numérico reducido de 12 teclas (fig. 8.16). Está formado por 10 teclas asociadas a las cifras del 0 al 9, y dos teclas disponibles para cada aplicación, que se rotulan y emplean según sea ésta. Existen leyendas estándar tales como (+) y «−», o «*» y «#», o «SEND» y «CLEAR»; aun cuando lo más usual es dejarlas en blanco o utilizar distintos colores identificativos. Su aplicación más usual (aparte de la telefonía) reside en equipos de control numérico y terminales simples de entrada de datos.

Una variante muy común del numérico reducido de 12 teclas, es el homónimo de 16 (fig. 8.17) que usualmente mantiene las 10 teclas correspondientes a los dígitos decimales, asignando las 6 restantes, bien sea a funciones específicas de la aplicación, bien sea a codificar los 6 últimos dígitos en numeración hexadecimal. Sus aplicaciones, aparte de las ya mencionadas en el párrafo anterior, incluyen control de microprocesadores, sistemas de seguridad, verificación de tarjetas de crédito, etc.

Entre los teclados alfanuméricos, dado el notable incremento en el número de teclas, las variedades existentes son

7	8	9
4	5	6
1	2	3
*	0	#

Figura 8.16 Teclado numérico reducido de 12 teclas

7	8	9	A
4	5	6	B
1	2	3	C
F	0	E	D

Figura 8.17 Teclado numérico reducido de 16 teclas

Figura 8.18 Teclado típico de máquina eléctrica de escribir

innumerables. Quizás el tipo más común (al menos entre los fabricantes de teclado) es el equivalente a las máquinas de escribir tipo «Selectric» (fig. 8.18) formado por 51 teclas, de las cuales 26 son alfabéticas, 10 numéricas, 8 caracteres especiales, 4 funciones y 3 control de conmutación mayúsculas/minúsculas.

Analizando la distribución de las teclas, puede observarse que no están distribuidas regularmente en sentido vertical (fig. 8.19), sino que existe un desplazamiento («offset») entre filas consecutivas. Este desplazamiento es una herencia de las máquinas de escribir mecánicas, en las que cada tecla estaba asociada a una palanca que impulsaba al tipo por su extremo; dado el espaciamiento de los tipos, se establecía una distribución uniforme de las palancas, con lo que no podían disponerse las teclas en forma cartesiana. Debido a que gran número de operadores provienen del área mecanográfica, se respeta dicha distribución con lo cual se facilita extraordinariamente la familiarización con nuevos equipos.

Las peculiaridades lingüísticas propias de cada idioma introducen una nueva dificultad, puesto que aparecen o desaparecen caracteres (como ejemplo, recuérdense la «Ç» catalana, o la «Ñ» castellana), en estos casos conviene situar

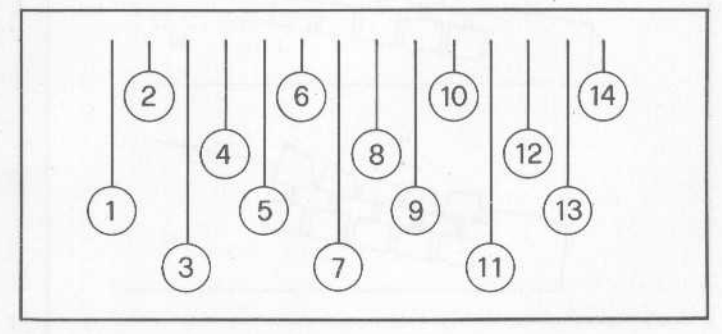


Figura 8.19 Desplazamiento (offset) entre filas



Figura 8.20 Teclado del terminal T-1800

los nuevos grafismos en las posiciones más usuales determinadas por la costumbre mecanográfica.

A menudo se asocia un teclado numérico reducido a la derecha del alfanumérico general, para facilitar la introducción de datos numéricos, mucho más frecuentes que los alfabéticos.

Un ejemplo de realización que combina los puntos antes citados, añadiendo funciones de control específicas de un periférico, puede ilustrar la situación (fig. 8.20). Este teclado forma parte de un terminal de pantalla en TRC diseñado y fabricado en España. Consta de 48 teclas alfabéticas, 11 numéricas y 15 de funciones, distinguiéndose estas últimas por ser de color oscuro. El teclado numérico reducido no comporta el desplazamiento entre filas, sino que adopta la estructura ortogonal propia de las máquinas calculadoras. Obsérvense como puntos distintivos, la presencia de la «Ñ», la tecla del «0» numérico más ancha, la tecla de mayúsculas duplicada a ambos lados de la barra espaciadora, el bloqueo de mayúsculas por la tecla «SHIFT LOCK» y las 5 teclas grafiadas con flechas. Estas últimas corresponden a funciones específicas de la pantalla.

Uno de los teclados de mayor difusión mundial es el popularizado por IBM en su PC (fig. 8.21). Los comentarios del párrafo anterior respecto a los grupos y la distribución de teclas son de plena aplicación, mereciendo resaltar los siguientes aspectos: las funciones de cursor (flechas) comparten las teclas numéricas, aparece un grupo de teclas de función programables (F1 a F10), y como caso particular se ha dibujado la distribución y grafismos de la versión española, en la que se pueden observar los diversos caracteres propios. En este punto cabe mencionar que la presente legislación exige que los teclados fabricados, importados y vendidos en España incorporen obligatoriamente la letra «Ñ».

Recientemente, la compañía IBM ha hecho evolucionar el teclado de su PC (fig. 8.22) de modo que se adapte mejor a aplicaciones de tipo profesional, aproximándose simultáneamente a la distribución de otro de los teclados más populares a nivel mundial, el del terminal VT-220 de DEC. Como puede observarse en la figura 8.22, este teclado consta de 101 teclas repartidas en 5 grupos: alfanumérico, funciones específicas, cursores, numérico y funciones programables.

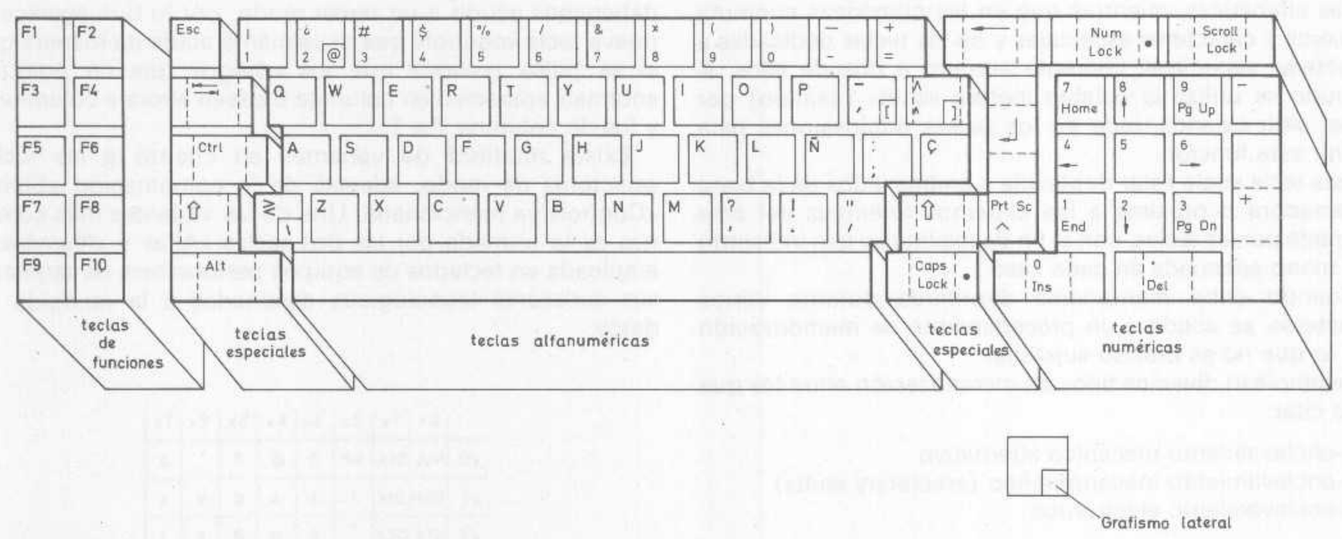


Figura 8.21 Teclado PC español

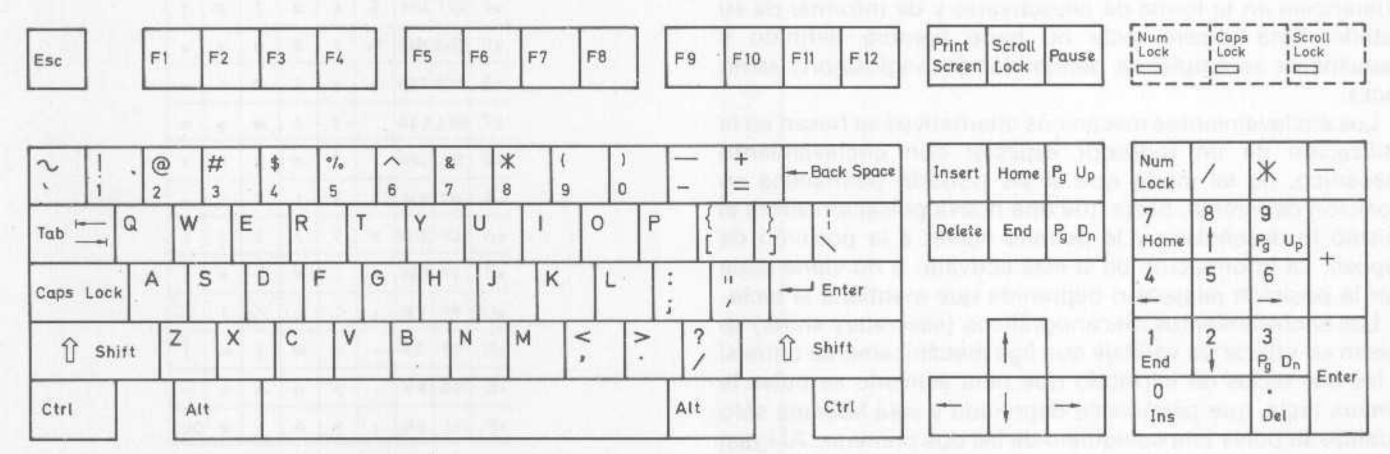


Figura 8.22 Teclado PC mejorado (versión básica)

8.6 MODOS

Un teclado mono-modo genera solamente un código por cada tecla. Los teclados de modos múltiples generan dos o más códigos por cada tecla en función de como se utilicen las teclas selectivas. El número de códigos generados por la depresión de una tecla es definido como el número de modos de un teclado.

Los teclados mono-modo son preferidos cuando el número de códigos es reducidos; los de dos modos se emplean para disponer de mayúsculas y minúsculas alfabéticas.

Los conjuntos de modos múltiples reducen la productividad del operador, puesto que debe utilizarse una mano para activar las teclas selectivas. Su ventaja consiste en la importante reducción en el número de teclas que lleva aparejado su empleo, siendo este factor primordial cuando no se exige elevada velocidad a la entrada de datos.

Los códigos estándar tales como ASCII o EBCDIC son particularmente adecuados al empleo de múltiples modos, puesto que por su propia estructura, se obtienen los códigos de los diversos modos de cada tecla con sólo modificar uno o dos bits.

Cuando un teclado debe trabajar en dos modos (mayúsculas/minúsculas) se utiliza una sola tecla para seleccionarlos. Esta tecla no tiene una denominación definida en castellano, por lo que a menudo se denomina como «tecla de mayúsculas», aún cuando sólo cumple la función de conmutar los códigos de mayúsculas y minúsculas en las teclas alfabéticas, mientras que en las numéricas conmuta números y caracteres especiales y en las teclas dedicadas a caracteres especiales conmuta entre sí a dos de ellos. A menudo se utiliza la palabra inglesa «shift» (cambio) por haber sido estandarizada en los países anglosajones para definir esta función.

Esta tecla suele estar duplicada a ambos lados de la barra espaciadora o próxima a los extremos inferiores del área ocupada por las teclas, con el fin de facilitar su uso indistinto a la mano adecuada en cada caso.

Cuando debe mantenerse deprimida durante varios caracteres, se acude a un procedimiento de memorización con lo que no es preciso sujetarla.

Se emplean diversos tipos de memorización entre los que cabe citar:

—enclavamiento mecánico alternativo
—enclavamiento mecanográfico («secretary shift»)
—enclavamiento electrónico

Todos estos procedimientos comparten el hecho de que para ser activados es preciso pulsar una tercera tecla, y se diferencian en la forma de desactivarse y de informar de su estado. Esta tercera tecla no tiene nombre definido y usualmente se emplea la denominación anglosajona «shift lock».

Los enclavamientos mecánicos alternativos se basan en la utilización de un pulsador especial con enclavamiento mecánico, de tal modo que al ser pulsado permanece en posición deprimida, hasta que una nueva pulsación sobre el mismo lo desenclava y le permite volver a la posición de reposo. La información de si está activado o no viene dada por la posición relajada o deprimida que mantiene la tecla.

Los enclavamientos mecanográficos («secretary shift») se basan en utilizar un varillaje que liga mecánicamente entre sí a las tres teclas de tal modo que para activarlo se pulsa la tercera tecla, que permanece deprimida y será liberada sólo cuando se pulse una cualquiera de las dos primeras. Al igual que en el caso anterior, la información relativa a si está activado o no, proviene de la posición de la tecla «shift lock».

Los enclavamientos electrónicos están ganando popularidad a medida que las técnicas de integración permiten aumentar en gran manera la cantidad de lógica asociada a cada periférico. Las tres teclas son normales tanto eléctrica como mecánicamente, con la única opción de asociar un piloto señalizador a la tecla «shift lock» (lo que en algunos casos modifica mecánicamente dicha tecla). Su funcionamiento está basado en asociar un biestable que se activa al pulsar la tecla de enclavamiento y se desactiva al pulsar cualquiera de las tres (algunos sistemas no son liberados por la pulsación repetida sobre la tercera). La información relativa a la selección o no del 2.° modo de codificación no viene dada en este caso por la posición de ninguna tecla sino por el estado de un flip-flop electrónico que suele visualizarse a través del piloto antes citado.

Cuando se desean 3 o 4 modos, es preciso disponer de otra función de selección distinta de «shift». Dado el reparto de caracteres de la tabla ASCII (fig. 8.23) normalmente empleada, puede observarse que en posición «normal» al pulsar las teclas se generan los códigos correspondientes a las columnas 3, 6 y 7 (excepción hecha de los caracteres especiales que se rigen por el uso); si se selecciona «shift», se generan respectivamente los códigos de las columnas 2, 4 y 5 (en ocasiones se utiliza «shift» inverso, de modo que en reposo genera mayúsculas y activado, minúsculas). Si deseamos generar algún código de las columnas 0 y 1 (correspondientes a controles de periféricos y transmisión) deberemos acudir a un tercer modo, por lo que aparece la nueva tecla «control» que usualmente actúa de manera que si se pulsa provoca que los códigos que en posición «normal» aparecían en columna 6 pasen ahora a columna 0, y los de columna 7 a 1.

Existe multitud de variantes en cuanto a las teclas selectoras de modo, además de la combinación «Shift»-«Control» ya mencionada. Una de las variantes más comunes es la formada por las dos teclas «Alfa» y «Numérico» empleada en teclados de equipos perforadores de tarjetas y sus sucesores tecnológicos destinados a la recogida de datos.

	0x	1x	2x	3x	4x	5x	6x	7x
x0	NUL	DLE	SP	0	@	P	`	p
x1	SOH	DC1	!	1	A	Q	a	q
x2	STX	DC2	"	2	B	R	b	r
x3	ETX	DC3	#	3	C	S	c	s
x4	EOT	DC4	$	4	D	T	d	t
x5	ENQ	NAK	%	5	E	U	e	u
x6	ACK	SYN	&	6	F	V	f	v
x7	BEL	ETB	'	7	G	W	g	w
x8	BS	CAN	(	8	H	X	h	x
x9	HT	EM	)	9	I	Y	i	y
xA	LF	SUB	*	:	J	Z	j	z
xB	VT	ESC	+	;	K	[	k	{
xC	FF	FS	,	<	L	\ / Ñ	l	¦ / ñ
xD	CR	GS	–	=	M	]	m	}
xE	SO	RS	.	>	N	∧	n	~
xF	SI	US	/	?	O	_	o	DEL

Figura 8.23 Código ASCII

En recientes versiones de teclados destinados a terminales y ordenadores personales aparecen teclas selectoras de modo, tales como:

— CAPS LOCK (enclavamiento de mayúsculas), de funcionamiento parecido a la tecla «shift lock» antes descrita, pero con la diferencia sustancial que sólo fuerza la segunda modalidad (mayúsculas) en las teclas alfabéticas, no actuando sobre el resto del teclado, de modo que si se desea la segunda modalidad en el resto de teclas debe pulsarse la tecla convencional de cambio («shift»).

— NUM LOCK (enclavamiento numérico), que permite compartir las teclas numéricas con otras funciones, tales como: controles de cursor, funciones programables, etc. Usualmente, esta tecla es de acción alternativa adoptando el estado numérico o el de función tras cada pulsación.

— ALT (alternativo), modalidad adicional a las teclas «shift» o «control» ya mencionadas.

8.7 DEPRESION SIMULTANEA DE VARIAS TECLAS

Un problema común a cualquier tipo de teclado es el que se provoca si se pulsan dos o más teclas simultáneamente. Si no se toma ninguna precaución, lo más común es que se provoque una suma inclusiva de bits dando lugar a la generación de un tercer código que no corresponde a ninguna de las dos teclas pulsadas, perdiendo asimismo la información correspondiente a estas últimas.

Aunque pueda parecer que este es un problema de operatoria ajena a los equipos, la tecnología ha desarrollado una serie de soluciones que permiten soslayar los defectos humanos de manipulación. Las soluciones más usuales son las siguientes (fig. 8.24):

—sobrepulsación de 2 teclas
—inhibición de *N* teclas
—sobrepulsación de *N* teclas.

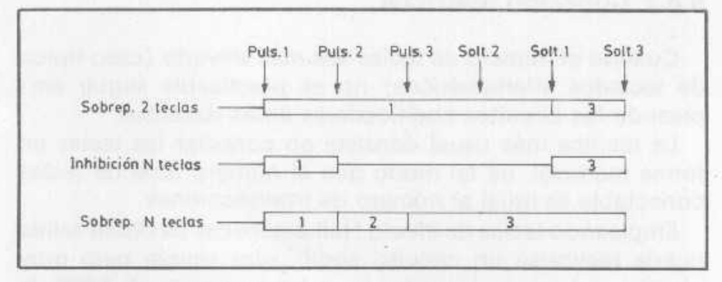


Figura 8.24 Depresión simultánea de varias teclas

8.7.1 Sobrepulsación de dos teclas

(En inglés: «2-Key rollover».) Cuando se pulsan varias teclas simultáneamente, sólo se transmite el código asociado a la primera, quedando la segunda y consecutivas bloqueadas hasta liberar la primera.

Si una segunda tecla fue pulsada tras la primera y liberada antes que ésta, no queda registrada, perdiéndose su información.

Este método garantiza que no aparezcan códigos erróneos, pero no impide pérdida de información.

8.7.2 Inhibición de *N* teclas

(En inglés: «N-Key lockout».) Cuando se pulsan varias teclas simultáneamente no se generan códigos a la salida. Cuando una sola tecla está deprimida, el teclado genera su código, pero cuando se pulsa una segunda tecla mientras la primera permanece activa, el teclado no generará ningún código mientras no se libera la primera. Una vez está liberada, el código correspondiente a la segunda aparecerá a la salida. Por tanto si se pulsan *N* teclas simultáneamente, permanecerá inhibida la codificación hasta que todas las teclas regresen a la posición de reposo, excepto una.

Este procedimiento es muy similar al de sobrepulsación de dos teclas, diferenciándose por el hecho de que en aquél durante la pulsación múltiple se dispone del código de la primera tecla; mientras que en éste, durante la pulsación múltiple, la salida permanece inhibida.

Merece el mismo comentario respecto a posibles pérdidas de información.

8.7.3 Sobrepulsación de *N* teclas

(En inglés: «N-Key rollover».) Cuando se pulsa una tecla, se genera su código correspondiente. Si la primera tecla permanece deprimida mientras se pulsa una segunda, se generará la salida correspondiente a la segunda tecla.

Si se pulsa una tercera tecla mientras las dos primeras (o alguna de ellas) están todavía activadas, se genera el código correspondiente a esta tercera tecla.

En un caso extremo, todas las teclas del teclado excepto una pueden ser pulsadas; cuando se activa la última tecla, se generará su código asociado.

Este método se encuentra comúnmente en máquinas eléctricas de escribir, donde ha demostrado su virtud de poder incrementar notablemente la velocidad de tecleo sin generación de errores, ni pérdidas de información.

Generalmente se acepta que los procedimientos de «inhibición de *N* teclas» o «sobrepulsación de dos teclas» son suficientes cuando aparece una indicación visual, tal como iluminación de una pantalla de TRC, impresión sobre papel o similares. El procedimiento de «sobrepulsación de *N* teclas», deseable en todos los casos, es absolutamente necesario cuando no se dispone de información visual asociada al teclado.

8.8 CODIFICACION

Como señales de salida de un teclado pueden utilizarse las conexiones correspondientes a todos y cada uno de los distintos conjuntos tecla-pulsador que lo constituyen. Esto puede ser válido para teclados simples formados por un reducido número de teclas; pero es claramente engorroso si el número total de teclas supera ciertos umbrales.

A título de ejemplo considérese que un teclado mínimo de 12 teclas; no es posible procesarlo mediante una sola palabra en sistemas basados en procesadores de 8 bits (aplíquese lo anterior a teclados alfanuméricos de más de 70 teclas).

Evidentemente se han buscado soluciones mucho más efectivas que han sido centradas en la codificación de los datos de salida.

Esta codificación consiste en numerar de forma binaria cada uno de los distintos códigos emitidos por el teclado, de tal modo que el número total de bits precisos para expresar cualquier código no supera los umbrales de maniobrabilidad.

La codificación más usual para teclados numéricos reducidos es la hexadecimal (o su subconjunto BCD si nos

limitamos a dígitos decimales); en el caso de teclados alfanuméricos se amplia la codificación, siendo el código más usual el ASCII (fig. 8.23) de 6 o 7 bits según sea reducido o completo, o bien el EBCDIC de 8 bits.

Por circuitos codificadores de un teclado, no solamente se entienden los circuitos precisos para comprimir el número de conexiones, sino además el resto de electrónica asociada a teclas y pulsadores; esto incluye, naturalmente, los circuitos destinados a generar las variantes asociadas a cada modo y los destinados a prevenir las pulsaciones simultáneas, antes mencionadas.

8.8.1 Rebote de contactos

Un factor a tener presente en los circuitos codificadores es la presencia de rebotes en los contactos.

Cuando se cierra un interruptor, los contactos no se detienen inmediatamente, sino que dado que en el momento del establecimiento del contacto la parte móvil se desplazaba a cierta velocidad, esta energía se restituye elásticamente, provocando una oscilación mecánica. La frecuencia de estos rebotes y el tiempo de amortiguación varían en función de las características propias del tipo de interruptor considerado. El tiempo de estabilización varía desde cero en los interruptores sin contactos, hasta 0, 5 a 1 ms en tipos más económicos. Los efectos de los rebotes en las salidas codificadas pueden ser eliminados introduciendo un retardo adecuado en la señal de validación.

Como ejemplo de lo dicho anteriormente se ilustra la codificación de un teclado de 10 teclas (fig. 8.25). Consta de 10 pulsadores, un codificador prioritario del tipo 74147 que entrega directamente la codificación BCD y un grupo de puertas para generar la señal de validación. La puerta 1 (representada como «Y» de 5 entradas; en la práctica sería una «Y» de 8, o bien una de 4 y otra de 2) suma lógicamente las salidas de los 4 bits de datos y de la tecla del «0», de modo que su salida baja a nivel 0 cuando es deprimida cualquier tecla. El diferenciador digital formado por las puertas 2 y 3, así como la constante *RC*, determina la generación de un impulso de duración *t*, superior al tiempo máximo de rebotes, de tal modo que durante el flanco de subida de la señal de «validación» los datos serán estables.

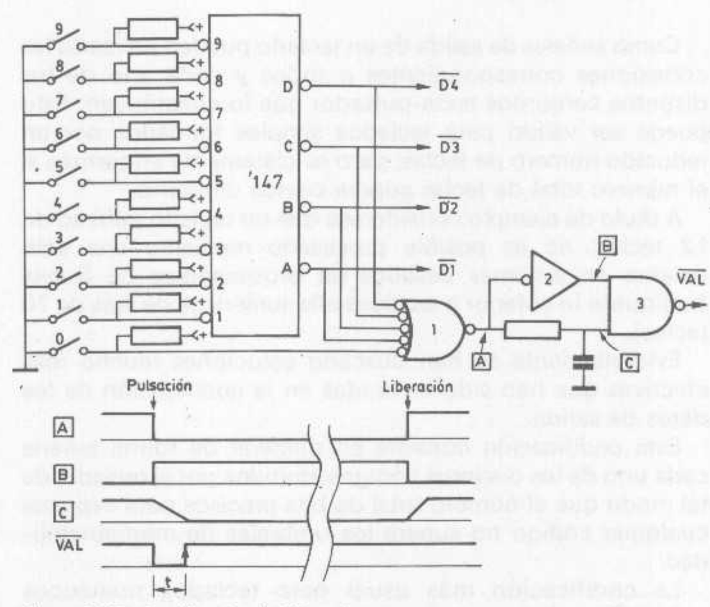


Figura 8.25 Codificador de 10 teclas

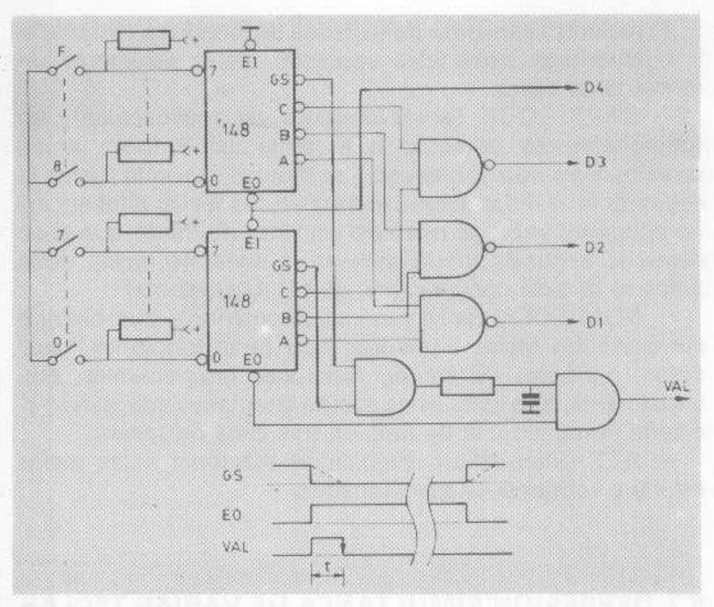


Figura 8.26 Codificador de 16 teclas

Una ampliación inmediata es la precisa para generar los 16 códigos hexadecimales (fig. 8.26). En este caso, en lugar de utilizar el codificador '147, se empleará el '148, que permite 8 entradas con salida en 3 bits, más las señales GS, EI y EO para encadenado y validación (ver su descripción en el capítulo de esta misma obra, dedicado a «E/S en paralelo»). Para permitir la conexión de las 16 teclas, deberán utilizarse dos circuitos integrados encadenados para transmitir las prioridades de tal modo que si se pulsan simultáneamente dos o más teclas, solamente aparecerá la codificación correspondiente a la tecla de numeración más alta entre las pulsadas, impidiendo por tanto mezclas incorrectas. De forma similar a la ya aplicada para el teclado de 10 teclas, puede generarse un impulso de validación que permita eliminar rebotes.

8.8.2 Conexión matricial

Cuando el número de teclas sea más elevado (caso típico de teclados alfanuméricos) no es practicable seguir empleando los circuitos codificadores antes descritos.

La técnica más usual consiste en conectar las teclas en forma matricial, de tal modo que el número total de teclas conectable es igual al número de intersecciones.

Empleando teclas de efecto Hall dinámicas de doble salida puede realizarse un circuito codificador simple pero muy efectivo, tal como el empleado en los terminales T-1800 de Telesincro (fig. 8.27).

Este teclado permite la conexión de hasta 128 teclas repartidas en una matriz de 8 filas por 16 columnas.

Mediante tres codificadores prioritarios del tipo '148 se obtiene un código de 7 bits, que no corresponde necesariamente a una codificación estándar tal como la ASCII. Con el fin de adaptar la codificación a la norma deseada, así como para obtener un teclado de 4 modos se inyectan los 7 bits antes obtenidos, junto con las salidas de las teclas estáticas «Shift» y «Control» a una PROM de 512×8 bits, que dará lugar a la salida de 8 bits transcodificados apropiadamente. La señal de validación se obtiene de modo similar al ya mencionado en casos anteriores.

8.8.3 Exploración secuencial

El ejemplo ilustrado en el párrafo anterior no puede

Figura 8.27 Codificación del teclado del terminal T-1800

generalizarse, por cuanto a menudo los pulsadores empleados no disponen de las elevadas prestaciones del efecto Hall dinámico. En estos casos (que comprenden todos los pulsadores mecánicos así como los «reed») se acude como norma general a realizar los circuitos codificadores empleando técnicas de exploración secuencial.

Un circuito clásico es el ilustrado en la figura 8.28 basado en un contador de 7 bits, un multiplexor y un decodificador de 4 a 16. Las teclas codificadas forman una matriz en la que cada tecla conecta una salida del decodificador con una entrada del multiplexor. El decodificador está seleccionado

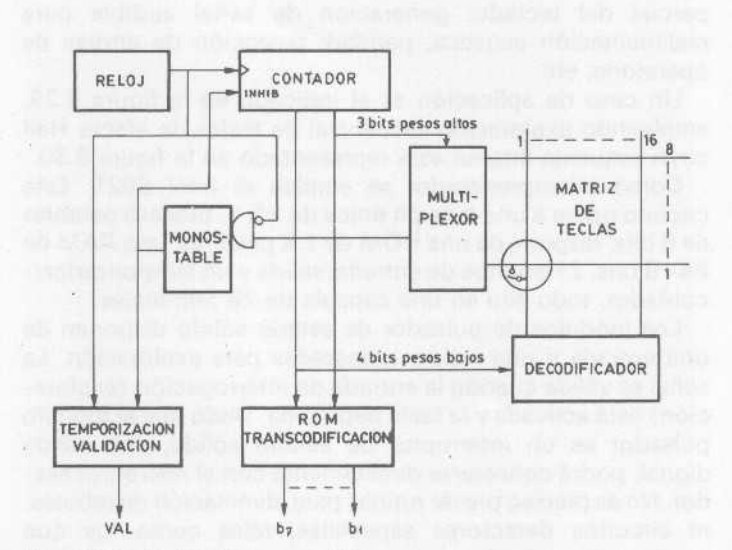


Figura 8.28 Exploración secuencial

por los 4 bits menos significativos y el multiplexor por los 3 más significativos del contador. Cuando se pulsa una tecla, se cierra una conexión, de tal modo que cuando el contador alcanza el código apropiado, el multiplexor conmuta a su salida y dispara un monostable redisparable, que detiene el contaje. El monostable se redispara continuamente mientras la tecla está pulsada.

Una PROM o circuitería equivalente realiza la transcodificación y adaptación de modos, aunque esta función puede ser realizada por el procesador mediante acceso a una tabla inserta en el microprograma.

Un esquema completo para codificar un teclado de hasta 128 teclas por el método de exploración secuencial resultaría bastante voluminoso si se realiza mediante integrados simples SSI y MSI, máxime si se incorporan circuitos complicados de selección de modos y protectores de sobrepulsaciones.

Estos casos han sido resueltos mediante circuitos integrados MOS-LSI, que realizan todas estas funciones. Es difícil dar una relación por cuanto cada fabricante dispone usualmente de varios modelos, algunos programables, no existiendo generalmente compatibilidad entre distintas marcas.

8.8.4 Codificación por microprocesador

Una técnica que se está expandiendo fuertemente consiste en la utilización de microcomputadores como elementos de lógica activa en la codificación de teclados. Esta técnica permite, con un mínimo de componentes, realizar funciones que hasta el presente raramente eran llevadas a cabo directamente por el teclado como periférico. Entre estas

Figura 8.29 Codificación por μP

funciones cabe mencionar: exploración secuencial; protección contra pulsaciones simultáneas; transcodificación; modos múltiples; selecciones de modo complejo; salidas en paralelo o en serie; memoria FIFO en caso de pulsación más rápida que el posible acceso por parte de la CPU; autorrepetición en teclas seleccionadas (se denomina autorrepetición al hecho de que si se mantiene deprimida una tecla, pasado un tiempo prudencial, entre 0,5 a 1 segundo, se repiten las validaciones del código asociado a un ritmo aproximado de 10 Hz); repetición por tecla «REPITE» independiente de ciertas teclas o funciones; autorización o inhibición total o parcial del teclado; generación de señal audible para realimentación acústica; paridad; detección de errores de operatoria; etc.

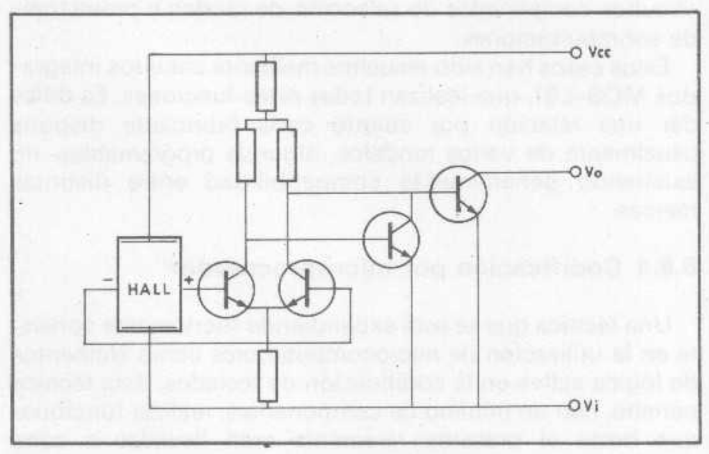


Figura 8.30 Tecla de efecto Hall para exploración secuencial

Un caso de aplicación es el indicado en la figura 8.29, empleando exploración secuencial de teclas de efecto Hall cuyo esquema interno está representado en la figura 8.30.

Como microprocesador se emplea el Intel 8021. Este circuito opera a una tensión única de +5 V, procesa palabras de 8 bits, dispone de una ROM de 1 K palabras, una RAM de 64 × 8 bits, 21 puertos de entrada/salida y un temporizador/-contador, todo ello en una cápsula de 28 terminales.

Los módulos de pulsador de estado sólido disponen de una entrada y una salida, preparadas para exploración. La señal es válida cuando la entrada de interrogación (exploración) está activada y la tecla deprimida. Dado que el módulo pulsador es un interruptor de estado sólido, con salida digital, podrá conectarse directamente con el microprocesador. No es preciso prever rutinas para eliminación de rebotes, ni circuitos detectores especiales, tales como los que precisan los pulsadores capacitivos o de núcleos de ferrita;

con todo ello se dispone de mayor espacio en la ROM para permitir incrementar sus prestaciones.

8.8.5 Doble codificación

El control del teclado puede efectuarse totalmente mediante un único microprocesador, tal como se ha visto en el apartado precedente, o bien distribuyendo el control entre el microprocesador localizado en el propio teclado y el situado en la unidad central. En este último caso pueden emplearse diversas soluciones, siendo la de doble codificación una de las más extendidas.

La técnica de doble codificación consiste en generar dos códigos por cada tecla, el primero en el momento de su depresión («make») y el segundo en su liberación («release»). Ambos códigos presentan una notable similitud; de hecho sus siete bits de menor peso son idénticos, diferenciándose en que el octavo bit es un «0» en la depresión y un «1» en la liberación. Los siete bits de peso inferior acostumbran a codificar de forma binaria la posición física de la tecla, siendo, por tanto, independientes del carácter o función asimilados a dicha tecla; como consecuencia, para obtener el código alfanumérico asociado a la tecla en cuestión, es preciso efectuar una transcodificación mediante el microprocesador de la unidad central. La tarea adicional que comporta esta rutina queda ampliamente compensada por la flexibilidad que se obtiene, puesto que cualquier tecla puede adoptar el significado que se desee, el cual puede ser modificado sin más que alterar la tabla de transcodificación activa. Asimismo, cualquier tecla puede adoptar la característica de tecla de modalidad (tal como: shift, control, alt., etc.) dado que el microprocesador de la unidad central es quien decide, en base a su microprogramación, la funcionalidad asignada a cada tecla. Asimismo, dado que la unidad central de control conoce en todo momento qué teclas están pulsadas (puesto que ya ha recibido el código de depresión y aún no el de liberación) pueden estructurarse secuencias complejas que comporten combinaciones simultáneas de múltiples teclas, o bien funciones tales como la autorrepetición «typematic» consistente en repetir a una frecuencia de 10 a 15 c/s el código de la tecla cuya pulsación haya sido mantenida un tiempo superior a 500-750 ms.

8.9 INTERFAZ

La conexión de un teclado completo a cualquier sistema informático, generalmente no plantea difíciles problemas por su simplicidad y baja velocidad de operación.

De hecho basta con tener presentes las señales que entrega, para deducir la interfaz precisa. El caso más usual es el constituido por 7 u 8 bits de datos (depende si se incluye la paridad) y una señal de validación. En casos particulares aparecen señales asociadas a funciones especiales, tales como repetición, ruptura de secuencia, solicitud de identificación, u otras.

Al igual que cualquier otro sistema de entrada/salida en paralelo, puede realizarse la interconexión por medio de interrupciones, o bien por «polling».

Para definir la técnica idónea deben tenerse presentes varios factores, entre los cuales hay que contar con que la frecuencia de aparición de información no será jamás muy rápida, pues procede de la acción física de las manos del operador, por lo que difícilmente existirá un período inferior a 50 ms entre las pulsaciones de mayor velocidad.

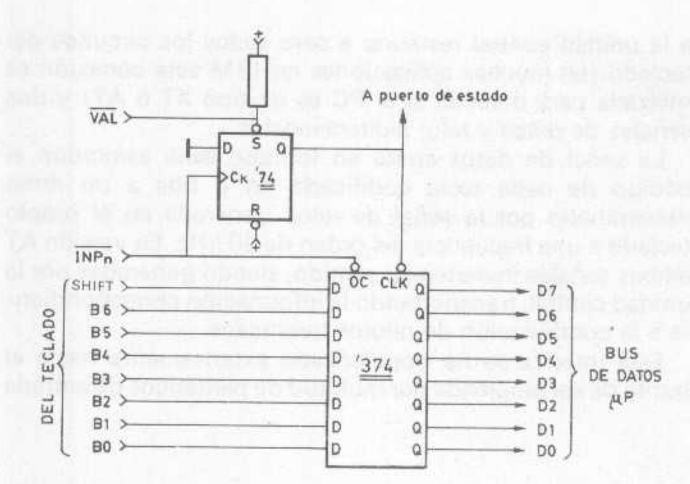


Figura 8.31 Interfaz «polling»

En sistemas que hagan uso extensivo de interrupciones puede ser ésta la solución más económica, aunque debe valorarse el tiempo de procesador destinado a atender al periférico, que evidentemente interrumpirá de forma asíncrona.

En muchos casos se prefiere actuar en polling, de tal modo que cuando el teclado dispone de una palabra, memoriza ésta en los circuitos de interfaz y activa un bit físico de estado. El procesador, durante la ejecución de las rutinas básicas, explora los bits de estado de ciertos periféricos (tal como el teclado) y dispara una rutina de entrada en momentos en que su atención no está destinada prioritariamente a atender fenómenos que exigen mayor velocidad de reacción. La figura 8.31 indica un ejemplo de realización.

No obstante lo expuesto en los párrafos anteriores, la interconexión más usual de teclado es la serie, por cuanto estos periféricos forman parte, normalmente, de terminales interactivos, en los que los datos transmitidos constituyen la información generada a través del teclado. Para ampliar estos casos, véase el capítulo dedicado a entrada/salida serie en esta misma obra.

8.9.1 Interfaz PC

Un caso particular de la interfaz serie, de muy amplia difusión, es la interfaz PC, denominada así por haber sido desarrollada por IBM en su PC.

A nivel físico, la interconexión se realiza a través de un conector DIN circular de 5 contactos a 180 grados, similar a los utilizados en aplicaciones de audio (fig. 8.32). Como se muestra en la misma figura, las cinco conexiones corresponden a: alimentación (+5 voltios y masa), «reset» que permite

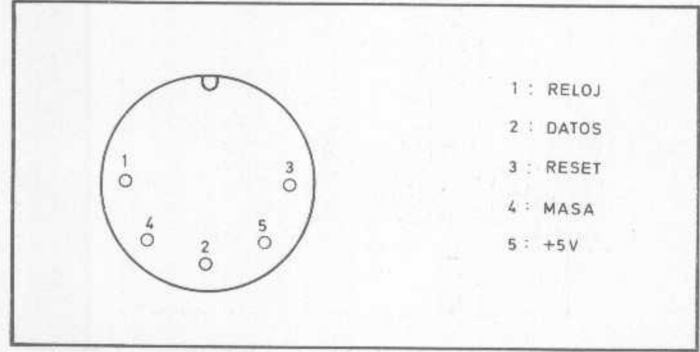


Figura 8.32 Interfaz PC

a la unidad central restaurar a cero todos los circuitos del teclado (en muchas aplicaciones no IBM esta conexión es utilizada para detectar si el PC es de tipo XT o AT) y dos señales de datos y reloj bidireccionales.

La señal de datos emite en formato serie asíncrono el código de cada tecla codificado en 8 bits a un ritmo determinado por la señal de reloj, generada en el propio teclado a una frecuencia del orden de 30 kHz. En versión AT ambas señales invierten su sentido, siendo generadas por la unidad central, transportando la información correspondiente a la conmutación de pilotos luminosos.

Esta interfaz se ha popularizado extensamente hasta el punto de ser empleada por multitud de periféricos de entrada manual, tales como ratones, tabletas digitalizadoras y similares.

BIBLIOGRAFIA

[1] Revista ELECTRONIC DESIGN
[2] Revista COMPUTER DESIGN
[3] Revista EDN
[4] Información técnica de MICROSWITCH (División de HONEYWELL)
[5] Información técnica de CLARE
[6] Información técnica de CHERRY
[7] Información técnica de KEYTRONIC
[8] Información técnica de RAFI

visualizadores TRC 9

José Bosch Solsona

9.1 IDEAS SOBRE EL VISUALIZADOR TRC

En un sistema microcomputador uno de los elementos más costosos es el terminal utilizado por el operador del sistema, ya que en el caso de utilizar un terminal teleimpresor tal como un teletipo, el precio de éste puede ser del orden de magnitud del precio del resto del sistema.

Una forma de solventar este problema consiste en utilizar como terminal un teclado y un visualizador de tubo de rayos catódicos (TRC), pudiendo ser éste el TRC del receptor de televisión del propio usuario.

Además, la reciente introducción de circuitos integrados LSI controladores de TRC hace que el diseño de un terminal de este tipo sea sencillo y su coste bajo.

En este capítulo se describe el funcionamiento de una unidad controladora de TRC, las posibles arquitecturas de diseño que se pueden utilizar, y los circuitos integrados controladores que existen actualmente.

Además se describe el funcionamiento de un visualizador TRC para posteriormente describir las principales funciones que debe realizar como terminal visualizador de datos de un sistema.

9.1.1 Funcionamiento del TRC

Tal como se ha mencionado antes, como visualizador TRC puede utilizarse el de un receptor de televisión comercial, cuyo diagrama de bloques aparece en la figura 9.1.

En la recepción normal de un programa de TV la señal captada por la antena pasa al sintonizador y después al amplificador de frecuencia intermedia, a la salida del cual está el detector de video; la señal de salida de este detector es la señal de video, con una amplitud aproximada de 2 voltios pico a pico. Esta señal es filtrada para eliminar la frecuencia portadora de video y después es introducida en el amplificador de video; dentro de este amplificador la señal sigue tres caminos: Por el primero de ellos llega la señal de video al TRC después de haber sido eliminada la frecuencia portadora de sonido y de haber sido amplificada; por el segundo, parte de la señal llega a la etapa de sonido y al altavoz, y por el tercero, los impulsos de sincronismo son separados de la señal de video y sirven para sincronizar la imagen en la pantalla.

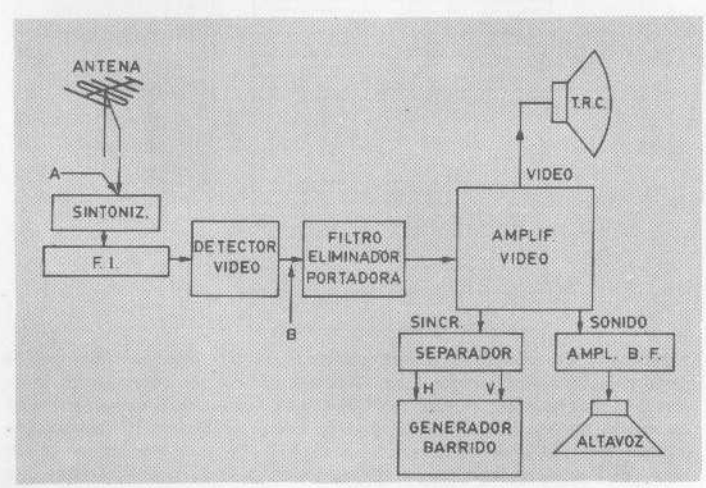


Figura 9.1 Diagrama de bloques de un receptor de TV

En el caso de utilizar un televisor como terminal de datos de un sistema microcomputador hay dos puntos por los que se puede introducir la señal proporcionada por el sistema al televisor: en la entrada de la antena, con una señal modulada convenientemente, o directamente a la salida del detector de video, bloqueando las etapas previas del televisor.

El primer sistema tiene la ventaja de que no debe alterarse para nada el receptor de televisión, pero tiene los inconvenientes de que debe modularse previamente la señal proporcionada por el sistema microcomputador con una señal portadora de la misma frecuencia que la de algún canal de televisión, y de que, debido al poco ancho de banda de los sintonizadores de radiofrecuencia y del amplificador de frecuencia intermedia (3,5 MHz aproximadamente), queda limitado el número de caracteres que pueden representarse en la pantalla.

Si la señal se introduce a la salida del detector de video se resuelven los dos problemas anteriores, pero entonces deberá modificarse algo el televisor ya que normalmente no lleva un conector externo para poder introducir una señal de video. Es asimismo aconsejable desconectar las etapas de alta frecuencia del televisor.

En el caso de utilizar como terminal un monitor digital de televisión, la solución a adoptar es obviamente la segunda al no disponer de las etapas de alta frecuencia. Estos monitores tienen la ventaja de que han sido diseñados especialmente para la visualización de caracteres alfanuméricos y el número de caracteres que se pueden visualizar con ellos es el doble o más que los de un televisor comercial.

9.1.2 Señales de control de un visualizador TRC

En un visualizador TRC dos generadores de rampa controlan el haz de electrones que inciden sobre la pantalla, modulado en intensidad por la señal de video.

La primera rampa controla el haz horizontalmente. Su período total es de 64 μs en el sistema europeo de TV (al que nos referiremos en lo sucesivo), del que aproximadamente la cuarta parte del tiempo se emplea para el retorno del haz.

La segunda rampa controla el haz verticalmente. Su período es de 20 ms, con lo que se definen 312 líneas horizontales en la pantalla (fig. 9.2).

Este tipo de barrido es el llamado directo y es el que suele utilizarse en los monitores de TV digitales. El otro tipo de barrido utilizado en la TV comercial es el de entrelazado, en el que cada imagen se representa mediante dos cuadros

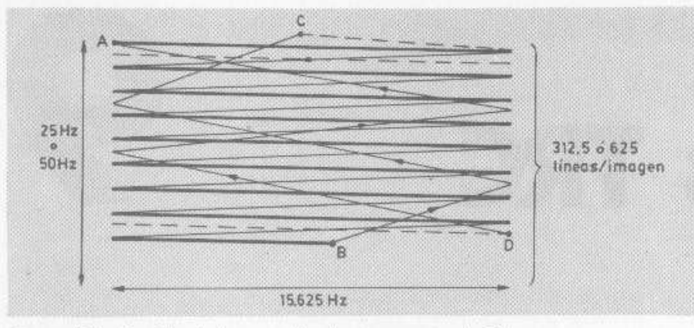


Figura 9.2 Barrido de la pantalla de un receptor de TV

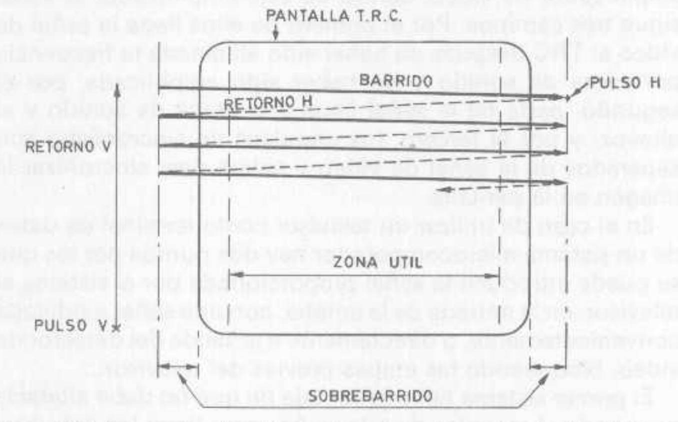


Figura 9.3 Forma de utilización de la pantalla de un visualizador TRC

consecutivos de 312 líneas y media cada uno, entrelazando las líneas del segundo cuadro con las del primero, lo que nos da un total de 625 líneas por imagen. En lo sucesivo solamente nos referiremos al sistema de barrido directo debido a que su resolución es suficiente y a que las señales de control se simplfican notablemente.

Las señales de control de un TRC son tres:

Sincronismo horizontal o señal H. Señal de impulsos cuyos flancos de subida provocan el retorno de derecha a izquierda de la pantalla del haz de electrones. La amplitud mínima de estos impulsos depende del monitor de TV empleado; la máxima, no es crítica. Durante el tiempo de retorno del haz éste se inhibe. Normalmente el barrido horizontal de la pantalla es mayor que el tamaño físico de la misma, y la longitud de una línea de caracteres es menor (fig. 9.3). El período del barrido horizontal es de 64 μs y la tolerancia es un 10 % como máximo.

Sincronismo vertical o señal V. Señal de impulsos cuyos flancos de subida provocan el retorno del haz de abajo a arriba de la pantalla. El haz de electrones también se inhibe durante este tiempo. Debe estar sincronizado con la señal *H*, y su período es de 20 ms.

Modulación de la intensidad del haz o señal Z. Es una señal normalmente binaria en el caso de visualizadores alfanuméricos y controla la intensidad del haz electrónico. Supondremos que su estado lógico alto provoca una intensidad máxima del haz, y su estado lógico bajo una intensidad mínima o nula.

La figura 9.4 nos muestra un ejemplo simplificado de las tres señales mencionadas anteriormente en el caso de representación de caracteres alfanuméricos. Si empleamos como monitor un receptor de TV e introducimos la señal después del detector de video, la señal adecuada es la señal *C* de la figura 9.4, que se obtiene sumando adecuadamente las señales *H*, *V*, y *Z*. Si la señal la introducimos por el conector de la antena, la señal *C* debe modularse mediante una portadora cuya frecuencia sea la de algún canal comercial de TV.

9.1.3 Funciones a realizar por el visualizador TRC

En el caso de utilizar un visualizador TRC como terminal de un sistema microcomputador, las funciones que debe realizar son las siguientes:

Representación de un texto alfanumérico. La finalidad principal de un visualizador TRC consiste en obtener una representación en la pantalla de un texto que tenemos almacenado en codificación binaria en una memoria digital.

La placa controladora del visualizador TRC deberá tener por tanto un generador de las dos señales de sincronismo *H* y *V* y un circuito controlado por dicho generador capaz de leer el texto almacenado en la memoria y transformarlo en la señal *Z*, que es la moduladora del haz electrónico.

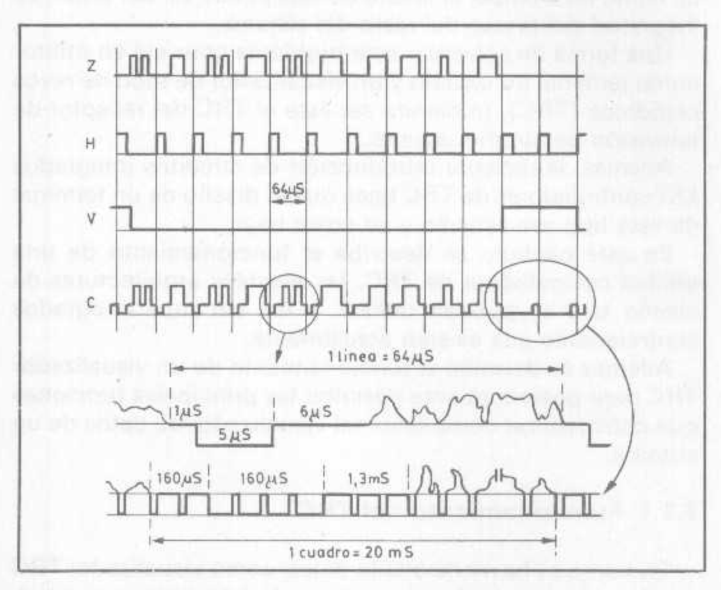


Figura 9.4 Señales de control de un visualizador TRC

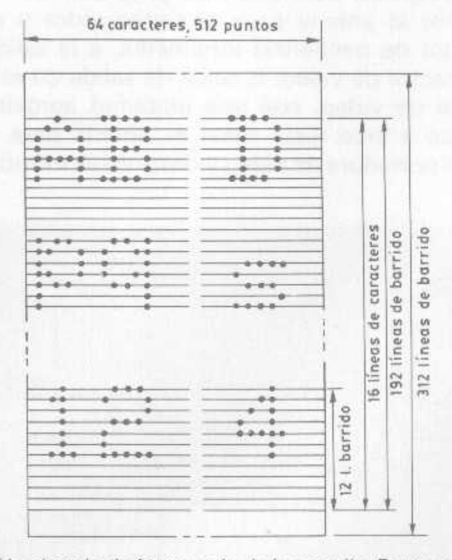


Figura 9.5 Un ejemplo de formateado de la pantalla. Formato: 16 líneas de carácteres: 64 caracteres por línea; 12 barridos/línea caracteres; Matriz 5 × 7, con 8 puntos/carácter; se precisan 512 puntos/línea para representar los 64 caracteres de una línea; se precisan 192 líneas de barrido horizontal para representar las 16 líneas de caracteres; si los 512 puntos se representan en 4/5 línea (51,2 μs) la frecuencia para los puntos es de 10 MHz precisándose una Bw del amplificador de video de 5 MHz

Si el sincronismo vertical tiene un período de 20 ms deberemos leer cada 20 ms el texto de la memoria y representarlo en la pantalla, con lo que se conseguirá una imagen estable.

Formateado de la pantalla (fig. 9.5). Un aspecto muy importante a tener en cuenta en el diseño es el formato que tendrá el texto en la pantalla: En qué zona de la pantalla habrá imagen, cuántas líneas de texto contendrá la pantalla, cuántos caracteres contendrá cada línea, cuántos puntos tendrá cada carácter, y cuántos caracteres diferentes podrán representarse.

Para que un visualizador TRC sea utilizable como terminal alfanumérico debe cumplir una serie de condiciones mínimas tales como las siguientes:

a) Representar 8 líneas de 16 caracteres cada una.
b) Utilizar formato de caracteres de 5×7.
c) Poder representar 64 tipos de caracteres diferentes.
d) El texto representado irá desfilando de abajo a arriba a medida de que se produzca nueva información para visualizar.

El formato mínimo de 8 líneas de 16 caracteres cada una puede corresponder a un visualizador de 12 pulgadas. Para visualizadores de grandes dimensiones (26 pulgadas, por ejemplo) puede llegarse a un formato de hasta 40 líneas de 84 caracteres cada una.

Formatos normalizados usados son los de la tabla 9.1, en la que además se indica el ancho de banda mínimo necesario para el amplificador de video. El barrido se supone del tipo directo, sin entrelazado. Todos estos formatos pueden adaptarse a la norma europea de TV de 625 líneas, de las que solamente son usadas 312.

La diferencia entre el número de líneas TV usadas y el número total de líneas de una imagen (312) corresponde a líneas que no se utilizan para la visualización y que están situadas en las partes superior e inferior de la imagen.

Funciones especiales. Si utilizamos como memoria del texto a visualizar parte de la memoria del sistema microcomputador, para modificar el texto que queremos visualizar basta modificar el contenido de la memoria bajo supervisión del mismo microprocesador. Si el sistema controlador del visualizador es autónomo e independiente del sistema microcomputador, la forma normal de funcionamiento es enviar carácter tras carácter al controlador del visualizador para que éste los incluya en su memoria local si los caracteres son representables, o realice ciertas funciones especiales si el carácter es una orden de control.

Ciertas funciones especiales son consecuencia inmediata de la llegada de un carácter de control, tal como el retorno de línea, salto de línea, etc. Otras funciones especiales deben realizarse automáticamente como resultado del estado actual del sistema, tales como el desplazamiento del cursor, el no incluir más caracteres en una línea cuando ésta está llena, etc.

Todas las funciones pueden realizarse mediante hardware especializado o mediante la gestión de la pantalla por un microprocesador dedicado especialmente a ello.

N.º líneas de caract.	*N.º caract. por línea*	*N.º total caracteres*	*N.º líneas TV usadas*	*Ancho de banda mínimo*
8	16	128	96	0,7 MHz
16	32	512	192	2,8 MHz
16	64	1024	192	5,6 MHz
24	80	1920	288	10,4 MHz

Tabla 9.1

Como funciones especiales principales merecen citarse las siguientes:

- Retorno de línea.
- Salto de línea.
- Tabulación vertical.
- Tabulación horizontal.
- Borrado de una línea.
- Borrado de un carácter.
- Borrado de todo el texto.
- Desplazamiento del cursor.
- Control de fondo.
- Escala de grises.
- Posibilidad de representación gráfica.
- Posibilidad de utilización de un «lápiz luminoso».

En los ejemplos de realizaciones prácticas que se dan en otro apartado se indicará cómo realizar algunas de estas funciones.

Si el visualizador es autónomo e independiente del sistema microcomputador deberá estar adaptado a las normas de transmisión de datos serie RS232 o bucle de corriente de 20 mA, o ser compatible con la lógica TTL en el caso de transmisión de datos en paralelo.

9.2 CIRCUITO CONTROLADOR DEL VISUALIZADOR TRC

La reproducción del texto sobre la pantalla del visualizador TRC se realiza en forma secuencial. Debe leerse el texto carácter a carácter, línea a línea, y una vez terminados la lectura total del texto y su representación sobre la pantalla, volver a empezar otra vez para que se obtenga una imagen estable ante el ojo humano.

Los caracteres se hallan representados en la memoria del sistema en código ASCII, código que una vez leído debe tratarse de tal forma que nos dé la información del carácter a representar en serie para poder visualizarlo en la pantalla.

Según lo anterior podemos ya deducir que habrá dos grandes bloques en un sistema controlador de un visualizador: la parte correspondiente a la generación de la señal de video y la parte correspondiente a la generación de las señales de control y de tiempo.

9.2.1 El generador de caracteres

Los caracteres alfanuméricos se representan sobre la pantalla del visualizador según un formato basado en una matriz de puntos. Dos son los formatos normalizados más comunes utilizados en la representación de caracteres alfanuméricos:

Formato 5×7: Se representa cada carácter mediante una serie de puntos distribuidos dentro de una matriz de 5 columnas por 7 filas (fig. 9.6). Un punto en la matriz representa un punto luminoso en la pantalla del visualizador, y su ausencia el efecto contrario.

Formato 7×9: La matriz en este caso tiene 7 columnas y 9 filas.

Ambos formatos son similares y presentan sus ventajas y sus inconvenientes. El formato 7×9 tiene una mayor resolución para representar los caracteres, pero la mayor cantidad de puntos en la matriz requiere un incremento del ancho de banda del amplificador de video para la representación de la misma cantidad de caracteres en una línea; en el caso de 64 caracteres por línea el ancho de banda requerido supera en varios megahertzios al ancho de banda de un receptor de TV normal. Por esta razón, el formato 7×9 suele

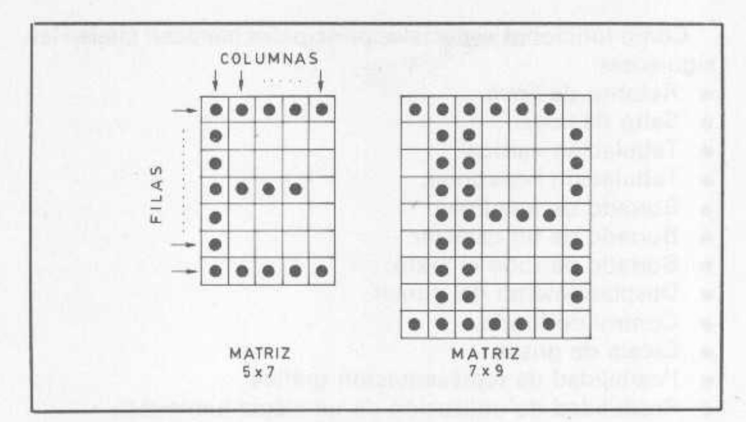


Figura 9.6 Formatos normalizados para la visualización de caracteres alfanuméricos

utilizarse solamente en los monitores digitales alfanuméricos.

Por contra, utilizando el formato 5×7 se puede obtener una resolución aceptable con un receptor normal de TV.

En un visualizador alfanumérico debemos poder representar por lo menos 64 tipos de caracteres diferentes. La información correspondiente a las matrices de puntos de todos estos caracteres suele estar almacenada en una memoria de semiconductores tipo ROM. Como parámetros de entrada del generador se precisa el código ASCII y el número de la fila correspondiente al carácter que se está visualizando; a la salida tenemos la información en paralelo de la fila de la matriz direccionada (fig. 9.7).

Los generadores de caracteres son manufacturados por la mayor parte de los fabricantes de semiconductores y es un componente de bajo coste. La forma práctica de utilizarlo está descrita en el apartado siguiente.

9.2.2 Generación de líneas de texto

Veamos ahora cómo representar una línea de texto formada por *n* caracteres. Si utilizamos una matriz de puntos del formato 5×7 precisamos como mínimo de 8 barridos horizontales de la pantalla del visualizador para su representación: un barrido para cada fila de la matriz de puntos más otro, por lo menos, para separar entre sí dos líneas de caracteres (fig. 9.5).

En el primer barrido representamos los *n* grupos de 5 puntos de los *n* caracteres de la línea, correspondientes a la primera fila de la matriz de puntos; en el segundo barrido se hace lo mismo para la segunda fila, y así hasta totalizar las 7 filas de la matriz de puntos.

Normalmente entre línea y línea de caracteres se intercala más de un barrido en blanco para obtener una mejor presentación visual del texto, con lo que el número de barridos por línea de caracteres será de 10, 12 o más.

Para representar completamente una línea de caracteres

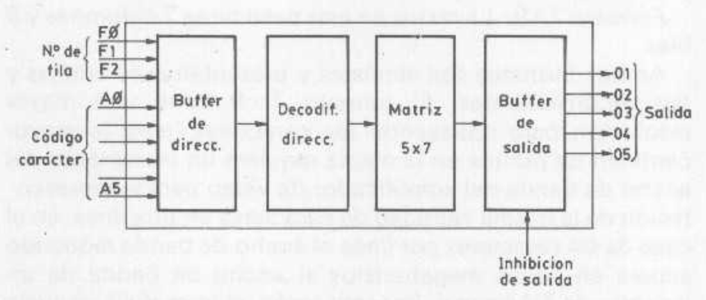


Figura 9.7 Diagrama de bloques de un generador de caracteres

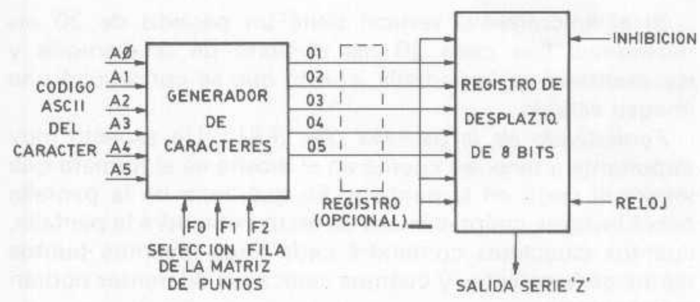


Figura 9.8 Serialización de la información a visualizar

alfanuméricos se van extrayendo los códigos ASCII de éstos de la memoria en forma secuencial tantas veces como filas tiene la matriz de puntos empleada, o sea, deben leerse los códigos de todos los caracteres en forma secuencial en cada barrido del haz electrónico. Se precisa, por tanto, un contador que nos indique cuál es el número del carácter de la línea que estamos visualizando en aquel momento, y de otro contador que nos indique cuál es el número de la fila de la matriz de puntos que debemos visualizar.

Como habrá también un contador de líneas de texto, en todo momento podremos saber la dirección absoluta de la posición de memoria que debemos leer en cada momento para obtener el código ASCII del carácter a visualizar. El código ASCII del carácter y el número de la fila de la matriz de puntos que se debe visualizar (fig. 9.7) son los parámetros de entrada del generador de caracteres. A la salida tenemos en paralelo la información referente a la fila de la matriz de puntos direccionada (cinco líneas de salida en una matriz 5×7), que se carga en paralelo en un registro de desplazamiento, gobernado mediante una señal de reloj de frecuencia igual a la frecuencia de visualización de los puntos sobre la pantalla. La salida serie de dicho registro es la señal de video que controla el haz electrónico.

Todo lo anterior podemos verlo de forma más clara en la figura 9.8. El código ASCII del carácter a representar se introduce en el generador de caracteres y mediante tres líneas de control adicionales se selecciona la fila de la matriz de puntos que debe visualizarse. La salida del generador de caracteres está conectada a un registro de desplazamiento de ocho bits, tres más que salidas tiene el generador y que corresponden a los espacios de separación entre dos caracteres consecutivos, por lo que están permanentemente a un nivel lógico bajo.

Nótese que para poder visualizar totalmente una línea de caracteres deben leerse los códigos de los mismos por lo menos tantas veces como filas tiene la matriz de puntos empleada. El tiempo de utilización de los buses es bastante grande. Una forma de reducir este tiempo solamente al tiempo de una sola lectura consiste en utilizar un registro de desplazamiento adicional que se carga con todos los códigos de los caracteres de la línea para hacer circular luego dicha información tantas veces como filas tiene la matriz de puntos. La desventaja de utilizar este sistema reside en que el registro de desplazamiento debe tener una longitud por lo menos igual al número de caracteres que caben en una línea (mínimo 16) y una longitud de palabra de por lo menos 5 bits. Puede emplearse una memoria FIFO como registro de desplazamiento.

9.2.3 Generación de un página de texto

La generación de una página de texto que contenga *n* líneas de caracteres se realiza repitiendo *n* veces el proceso

indicado en el apartado anterior. Se precisa, por tanto, un contador que nos indique en cualquier momento el número de línea visualizada.

La lectura de las líneas de texto en la memoria puede efectuarse de dos formas:

Lectura secuencial. El texto representado en la pantalla del visualizador está dispuesto exactamente de la misma forma que en la memoria del sistema. La dirección inicial de cada línea de texto se calcula a partir de la dirección inicial del texto y del estado del contador de líneas visualizadas.

Tiene la ventaja de su gran sencillez y de que es fácil de gobernar, y el inconveniente de que la modificación de una línea de carácteres implica muchas veces la modificación parcial o total del contenido de la memoria del texto. Asimismo, no puede alterarse fácilmente el orden en que se visualizan las líneas del texto.

Lectura aleatoria. El texto representado en la pantalla del visualizador no está dispuesto de la misma forma que en la memoria, sino que la dirección de cada línea se calcula a partir del contenido de una memoria auxiliar asociada al contador de líneas visualizadas; para modificar el orden de las líneas visualizadas basta modificar el contenido de dicha memoria.

Permite mayor flexibilidad en la utilización del visualizador, aunque se aumenta la complejidad en el diseño y en la gestión del mismo. En este caso resulta imprescindible que la gestión la lleve un microprocesador.

Deberá tenerse en cuenta el tiempo de retorno vertical del haz electrónico, durante el cual debe suspenderse la visualización, para reinicializarla en el momento adecuado poniendo el contador de líneas a cero, y repitiendo el proceso de visualización de una página de texto.

9.2.4 El generador de sincronismos. Registros asociados

El generador de sincronismos tiene por misión general las señales *H* y *V* de control del visualizador TRC y una serie de señales adicionales que nos controlen en tiempo los contadores y registros asociados al circuito de control del visualizador.

El generador de sincronismos está basado en un oscilador cuya frecuencia debe ser múltiplo a la vez de las frecuencias de línea y de cuadro del visualizador, y de la frecuencia de los puntos visualizados. En la norma europea de TV la frecuencia de cuadro (con entrelazado) es de 25 Hz y la de línea es de 15.625 Hz, con un tiempo de duración de la línea de 64 μs.

Veamos ahora cómo se calcula la frecuencia de oscilación fundamental del generador de sincronismos a partir del formato de una línea de caracteres y tomando como base la norma europea de TV. Supongamos que la línea es de 64 caracteres, con una resolución de 8 puntos por carácter, y que dicha línea ocupe solamente las 4/5 partes del tiempo de un barrido horizontal, dejando el resto como tiempo de retorno del haz electrónico. El período entre dos puntos será: T = (64 μs/barrido) × (4/5 usados) / (64 caract./línea) / (8 puntos/carácter) = 1/10 μs y la frecuencia del oscilador: $f = 1/T$ = 10 MHz.

El ancho de banda necesario del amplificador de video del visualizador para este caso sería de 5 MHz, la mitad de la frecuencia del oscilador.

A partir de la frecuencia base ya se pueden obtener las diferentes frecuencias necesarias:

f puntos = f osc.
f carácter = f osc./8
f línea = f osc./(64 × 5/4 × 8) = 15.625 Hz
f cuadro = f osc./(64 × 5/4 × 8 × 625) = 25 Hz

Como consideramos que no hay entrelazado, la frecuencia de cuadro será de 50 Hz.

La manera de construir el generador de sincronismos consiste en ir añadiendo en cascada a partir del oscilador base diversos divisores que nos vayan dando las frecuencias necesarias. Los contadores pueden realizarse a partir de las salidas de los divisores y de una pequeña parte adicional de puertas lógicas. La duración de los impulsos de sincronismo puede obtenerse de forma análoga.

Las señales de sincronismo *H* y *V* del visualizador TRC se obtienen a partir de las frecuencias de reloj de 15.625 Hz y 50 Hz. Mezclando adecuadamente estas dos señales con la que se obtiene a la salida del generador de caracteres la señal compuesta de video *C*.

Los contadores o registros asociados a la cadena de divisores nos proporcionan de forma implícita las coordenadas del haz electrónico sobre la pantalla en cualquier momento, y la dirección y coordenadas del carácter que se está visualizando. Como registros básicos podemos citar:

Indicadores de las coordenadas del haz electrónico

— Registro del número de línea de barrido.
— Registro de posición del haz dentro de la línea.

Indicadores de las coordenadas del carácter visualizado

— Registro del número de línea de caracteres.
— Registro del número de carácter en la línea.
— Registro del número de fila de la matriz de puntos.
— Registro del número de columna de la matriz.

La dirección absoluta en memoria del carácter visualizado se obtiene a partir de la dirección del primer carácter del texto a visualizar, que puede ser implícita o estar en un nuevo registro, del registro de línea de caracteres, y del registro de posición del carácter dentro de esta línea.

9.2.5 Sincronismo de una página de texto. El cursor

El cursor, representado normalmente en la pantalla del visualizador TRC mediante un pequeño guión, indica la posición en que se visualizará el próximo carácter que llegue al sistema visualizador, y sirve muchas veces para sincronizar la página de texto.

El cursor está situado normalmente dentro de la última línea de texto representada en la pantalla y se va desplazando de izquierda a derecha ante la llegada de nuevos caracteres. Sólo debe existir un cursor en una página de texto, utilizando para su representación en la memoria del texto un bit adicional para cada carácter y teniendo valor 1 lógico solamente el carácter al que está asociado el cursor.

Si se utiliza el cursor como sincronismo de página de texto, la creación de una nueva línea de texto no implica sino solamente el desplazamiento del cursor hacia el primer carácter de la línea que lógicamente se ha tenido que borrar para dar cabida a la nueva. El controlador del visualizador puede así calcular la dirección absoluta del primer carácter a visualizar a partir de las coordenadas del cursor.

El control del posicionamiento del cursor dentro de la memoria del texto debe efectuarse mediante un microprocesador, o mediante un circuito controlador especial. Parte de la gestión del cursor suele estar integrada en los CI controladores de visualizadores TRC.

En los sistemas en que no se utiliza el cursor como forma de sincronismo de página de texto, la dirección del primer carácter visualizado en la pantalla es siempre la misma en el

caso de utilizar memoria local o memoria común de página fija, o es introducida por el microprocesador cargando un registro de dirección inicial de página en el caso de disponer de varias páginas de texto.

9.2.6 Funciones adicionales del controlador del TRC

Hasta aquí se han explicado las funciones básicas e imprescindibles que debe efectuar un controlador de un visualizador TRC alfanumérico y las posibles formas de realizarlas. Existe además una serie de funciones adicionales, mencionadas en parte en el apartado 9.1.3, que pueden implementarse añadiendo circuitos especiales de control. En este apartado vamos a describir cómo se implantan algunas de las funciones que precisan una modificación del circuito controlador básico, sin tener en cuenta la representación tricromática ni las funciones que sólo precisan una gestión adicional de la memoria de texto mediante un μP.

Representación del fondo. Inversión del video. Hasta ahora hemos supuesto que un carácter se representa en la pantalla de color blanco sobre fondo negro. Esta representación se puede invertir si se invierte la señal Z de la figura 9.9. Para ello basta asociar un bit más a cada palabra de la memoria de texto que nos indique la forma de representación, sobre fondo negro o sobre fondo blanco.

A la salida del circuito del generador de la señal Z deberá añadirse un circuito adicional que invierta o no dicha señal de acuerdo con el valor del bit de fondo asociado al carácter que se está representando.

Puede realizarse la inversión del video de otras formas diferentes e igual de sencillas.

Escala de grises. Modulación de la intensidad. Para la modulación de la intensidad pueden utilizarse, por ejemplo, dos bits adicionales asociados a cada palabra de la memoria de texto que nos controlen un sistema analógico amplificador de la señal de salida Z del generador de caracteres. La señal Z será entonces analógica y capaz de modular en intensidad al haz electrónico.

Nótese que si queremos que el sistema sea capaz de efectuar nuevas funciones, la longitud de la palabra de la memoria de texto aumenta, dependiendo de la cantidad de funciones adicionales. Para un sistema básico 6 bits son suficientes; la longitud máxima puede llegar en algunos casos a 12 o 16 bits por palabra.

Representación gráfica. En el caso de querer utilizar el visualizador en forma gráfica en lugar de alfanumérica basta cambiar el generador de caracteres alfanuméricos por uno gráfico.

Cabe la posibilidad de tener representación alfanumérica

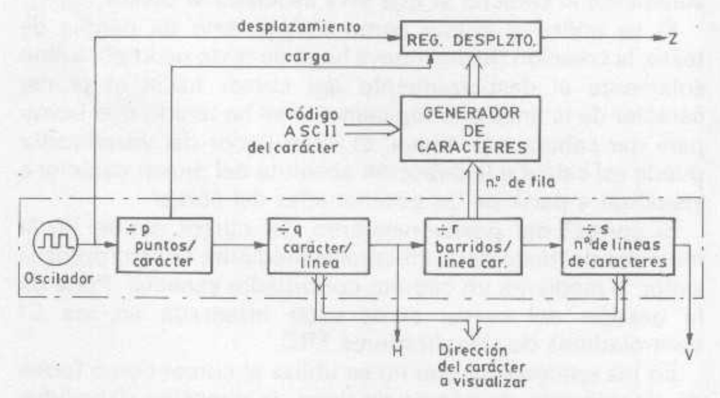


Figura 9.9 Diagrama de bloques del generador de sincronismo y su conexión al generador de caracteres

y gráfica simultánea dentro de la misma imagen si se utilizan simultáneamente los dos generadores, utilizando uno u otro en función de un nuevo bit asociado a cada palabra de la memoria del texto.

Puede incorporarse un «lápiz trazador» en el caso de representación gráfica para poder «dibujar» sobre la pantalla del visualizador, aunque en este caso la gestión pertinente es bastante compleja.

9.3 INTERCONEXION DEL CIRCUITO DE CONTROL CON EL μP

En los apartados anteriores hemos descrito el funcionamiento de un monitor digital de TV y los circuitos necesarios para poder utilizarlo como visualizador de caracteres alfanuméricos. No obstante nos falta aún un elemento que sirva como supervisor general de todo el sistema, capaz de proporcionar las órdenes de control necesarias y de hacer el pretratamiento de los caracteres visualizables.

Existen dos soluciones posibles para diseñar el supervisor general del visualizador TRC. La primera está realizada mediante circuitos lógicos discretos, y no vamos a hablar de ella ya que su forma de funcionamiento es muy rígida; la segunda solución incorpora un microprocesador capaz de controlar al visualizador TRC, y en este caso el pretratamiento de los caracteres a visualizar es muy sencillo. A partir de este momeno supondremos que el microprocesador es un elemento más del sistema visualizador.

Aparte del pretratamiento de los caracteres, que se verá posteriormente, aún nos falta analizar ciertos aspectos sobre la forma de interconectar el microprocesador con el circuito de control del visualizador, que es función de la forma en que queramos refrescar la imagen visualizada.

9.3.1 Refresco de la pantalla del visualizador

Para que la imagen visualizada sea estable ante los ojos del operador hemos visto anteriormente que es necesario refrescar la pantalla del visualizador periódicamente y a la frecuencia de cuadro.

El refresco de la pantalla consiste en reescribir en ella cada 20 ms aproximadamente todo el texto a visualizar, con lo que deberemos transferir periódicamente todos los códigos de los caracteres visualizables desde la memoria de texto al circuito de control.

Podría pensarse primero en separar completamente la memoria de texto de la memoria accesible por el microprocesador, pero entonces éste no podría operar sobre la misma y el uso del microprocesador sería innecesario. Si como memoria de texto se utiliza parte de la memoria direccionable por microprocesador, son posibles cuatro formas de intercambio de datos entre dicha memoria y el sistema de control del visualizador.

a) Mediante muestreo. El microprocesador puede estar tanteando o «testeando» continuamente una señal proporcionada por el sistema de control del visualizador, y cuando éste se lo indique, extrae el código de un carácter de la memoria de texto y se lo envía a través de un puerto de entrada/salida.

Este sistema es el más simple pero el menos adecuado, ya que sólo se consigue una velocidad de transferencia de unos 1000 octetos por segundo, y si tenemos en cuenta que el refresco de la pantalla debe realizarse cada 20 ms, el número máximo de caracteres visualizables será de 1000 B/s × 20 ms = 20 caracteres, cantidad muy pequeña.

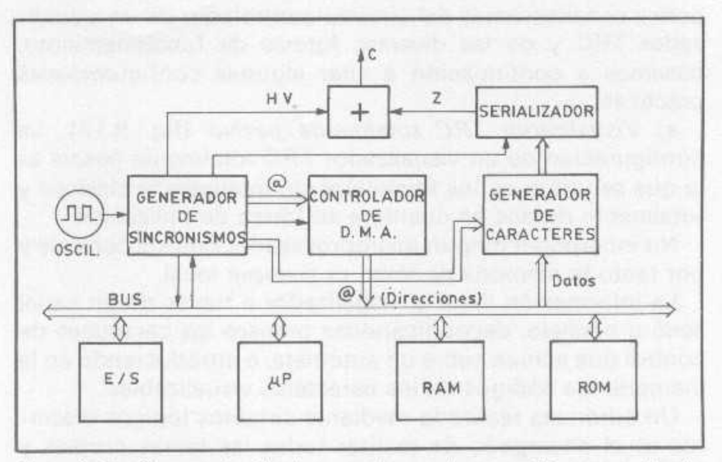


Figura 9.10 Sistema visualizador completo en el que se utiliza acceso directo a memoria (DMA)

Otro inconveniente es que el microprocesador precisa de todo su tiempo para llevar el control de la pantalla y no puede realizar otras tareas.

b) *Mediante interrupciones.* Sistemas análogo al anterior, pero con la diferencia de que los caracteres le son pedidos al microprocesador mediante una señal de interrupción. La velocidad de transferencia es superior y llega a los 10 kB/s, con unos 200 caracteres visualizables, cifra que sigue siendo pequeña.

c) *Mediante acceso directo a memoria (DMA)* (fig. 9.10). La velocidad de transferencia obtenida mediante este sistema es del orden de 100 kB/a 1 MB/s, con lo que se pueden representar más de 1000 caracteres en la pantalla.

Este sistema se utiliza normalmente y sólo exige para la transferencia de datos al visualizador un tiempo del orden del 5 % al 33 % del tiempo total, con lo que al microprocesador le queda suficiente tiempo para poder realizar otras tareas.

Para poder realizar este sistema, el circuito de control del visualizador debe ser capaz de controlar el bus del microprocesador durante la transferencia de la información.

La prioridad del DMA del visualizador debe ser lo más alta posible, ya que en el caso de que el microprocesador no pueda dejar el bus, la imagen de la pantalla fluctuaría al no poderse realizar un refresco o más. Este es el único inconveniente que presenta.

d) *Mediante acceso transparente a la memoria de texto* (fig. 9.11). En este sistema se multiplexan las direcciones de memoria que proporciona el controlador del visualizador con

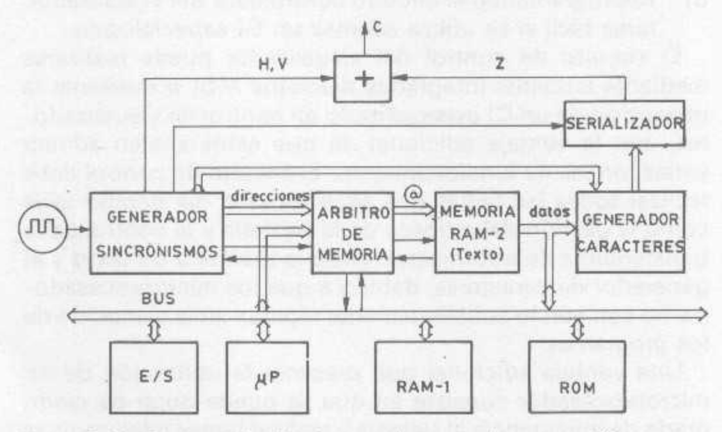


Figura 9.11 Sistema visualizador completo en el que se utiliza acceso transparente a la memoria de texto

las direcciones de memoria de texto proporcionadas por el microprocesador de tal forma que ambos puedan acceder a la memoria de texto de forma síncrona.

El acceso a la memoria por parte de uno de los dos sistemas resulta transparente para el otro y no hay pérdidas de tiempo por parte del microprocesador durante la transferencia de información. El único inconveniente es realizar el árbitro para el control del acceso a la memoria de texto, y la solución no es única ya que la realización es diferente para cada microprocesador.

Algunos CI especializados en control de visualizadores tienen incorporado este sistema, como se verá posteriormente.

9.3.2 Formas de gestión del refresco de la pantalla

En el sistema visualizador debe existir algún elemento encargado de dar las órdenes oportunas para realizar la tarea de transferencia de información desde la memoria de texto hasta el controlador del visualizador en el momento en que se deba refrescar la pantalla.

Esta tarea la puede realizar el microprocesador o el circuito de control. Veamos a continuación cuál de los dos sistemas nos proporciona una gestión más adecuada.

a) *Gestión mediante el microprocesador.* La gestión del refresco de la pantalla mediante el microprocesador resulta imposible en el caso de un formato de pantalla superior al de 8 líneas de caracteres por 16 caracteres/línea debido a que el tiempo que tardaría en realizar el programa de control es elevado en comparación con los tiempos de refresco de la pantalla.

La única forma de llevar a cabo la gestión mediante el microprocesador sería utilizar dos líneas de interrupciones que le indicasen cuándo comienza un cuadro y cuándo comienza una línea, y aún así no le quedaría tiempo para realizar las tareas asociadas a los caracteres de control de la pantalla ni el pretratamiento de los caracteres.

b) *Gestión mediante circuitos especializados.* Este es el único sistema práctico, y consiste en incluir la gestión del refresco como una tarea más del circuito de control del visualizador.

Si utilizamos DMA para la transferencia de información (fig. 9.10) debe diseñarse un autómata que sea capaz de pedir el bus al microprocesador cada vez que deba refrescarse la pantalla, generar las direcciones de la memoria de texto donde se encuentran los caracteres a visualizar, y extraer los códigos de los caracteres de dicha memoria. Posteriormente veremos un ejemplo práctico.

Si utilizamos la forma de acceso transparente, debemos ejecutar primero el árbitro para la memoria y luego el sistema que sea capaz de extraer los códigos de los caracteres (fig. 9.11).

Utilizando estos sistemas, el tiempo de ciclo del microprocesador ya no influye en la cantidad de caracteres visualizables y sólo pierde algo de su tiempo en el caso de utilizar DMA para la transferencia de información.

La utilización de CI especializados en el control de visualizadores nos simplifica notablemente el problema, ya que la gestión del refresco es una función que todos ellos llevan incorporado.

9.3.3 Formas de utilización de la memoria de texto

Se han indicado ya anteriormente algunas posibilidades de cómo puede estar organizado la memoria de texto en la que se encuentran los códigos de los caracteres a visualizar.

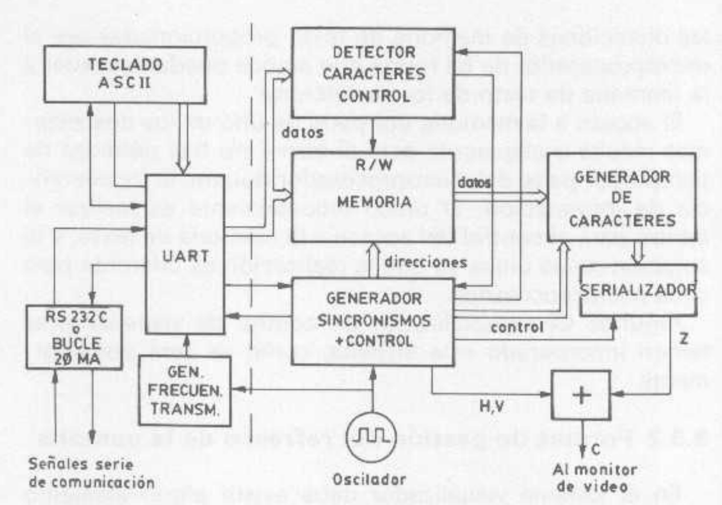


Figura 9.12 Diagrama de bloques de un terminal pasivo compuesto por visualizador TRC alfanumérico y un teclado ASCII

A continuación analizaremos las prestaciones de las configuraciones posibles.

a) Memoria de texto local. La memoria no es accesible por el microprocesador y es local en el sistema de control del visualizador (fig. 9.12).

Utilizan memoria de texto local los visualizadores que son totalmente pasivos y que no llevan un microprocesador de control, con lo que este sistema, aparte de su sencillez, adolece de todos los inconvenientes de no poder utilizar un microprocesador.

b) Página de texto fija y sin memoria local. Como memoria de texto se utiliza parte de la memoria accesible por el microprocesador (fig. 9.10 y 9.11).

La transferencia de información entre la memoria y el sistema de control del visualizador puede hacerse mediante DMA o por acceso transparente, y el microprocesador puede acceder a la misma para introducir o quitar caracteres, y para efectuar tareas asociadas a los caracteres de control.

La referencia alusiva a que la página de texto es fija se refiere a que la memoria de texto sólo puede almacenar igual número de códigos de caracteres que el número total de caracteres visualizables en la pantalla, y el sistema que gestiona el refresco de la pantalla tiene preestablecidas las direcciones inicial y final de la memoria de texto, sin posibilidad de poder acceder a otras zonas de memoria del microprocesador.

c) Página variable o introducida por el microprocesador. Se trata del mismo caso que antes, pero con la salvedad de que se dispone de varias páginas de texto sobre la memoria del microprocesador y bajo ciertas órdenes de control se selecciona qué página se va a visualizar. Al iniciarse un refresco de la pantalla, el controlador del visualizador pide al microprocesador la dirección de la página que debe refrescar mediante una interrupción.

Con este sistema se tiene la ventaja de que se puede manejar un texto mucho más extenso que el que se puede representar sobre la pantalla, y el inconveniente de que se le quita al microprocesador memoria en la que se podrían almacenar otros programas.

Los CI especializados en control de visualizadores están normalmente preparados para poder manejar varias páginas de texto.

9.3.4 Principales soluciones

Vistas ya las diferentes soluciones para cada una de las partes constituyentes del circuito controlador de un visualizador TRC y de las diversas formas de funcionamiento, pasamos a continuación a citar algunas configuraciones prácticas.

a) Visualizador TRC totalmente pasivo (fig. 9.12). La configuración de un visualizador TRC totalmente pasiva es la que se utiliza en los terminales sin inteligencia clásicos y totalmente rígidos en cuanto a su forma de utilización.

No incorporan ningún microprocesador que los controle y por tanto la memoria de texto es siempre local.

La información llega al visualizador a través de un canal serie o paralelo, decodificándose primero los caracteres de control que actúan sobre un autómata, e introduciendo en la memoria los códigos de los caracteres visualizables.

Un autómata realizado mediante circuitos lógicos discretos es el encargado de realizar todas las tareas propias y básicas de un visualizador TRC, desde la selección entre los caracteres de control y los visualizables, hasta la generación de las señales de sincronismo del visualizador.

La forma de utilización es rígida porque el autómata no es reprogramable y sólo tiene una forma de utilización fijada por el diseñador: formateado de la pantalla único, caracteres de control no ampliables, etc.

Existe un CI tipo LSI concebido para utilizarlo en terminales totalmente pasivos, el SFF 96364 de tecnología de fabricación NMOS, aparecido no hace mucho tiempo, y que permite la realización de terminales visualizadores de forma sencilla.

Algunos de los restantes CI especializados en control de visualizadores y concebidos para poder trabajar junto con un microprocesador también pueden utilizarse para la realización de terminales pasivos, pero tienen el inconveniente de que deben programarse con la ayuda de una memoria ROM cada vez que se ponen en marcha.

No es aconsejable actualmente realizar terminales visualizadores alfanuméricos mediante circuitos lógicos discretos o MSI.

b) Visualizador TRC que incorpore un microprocesador. Un sistema visualizador que incorpore un microprocesador tiene la ventaja de que el circuito de control propio del visualizador es más sencillo al haber un pretratamiento de la información a visualizar por parte del microprocesador.

La memoria de texto forma parte de la memoria del microprocesador y puede tener una configuración de varias páginas de texto, seleccionándose de forma fácil la página a visualizar.

El microprocesador no controla directamente al visualizador, sino que lo hace a través de dos caminos:

a) Modificando el contenido de la memoria de texto.
b) Reprogramando el circuito controlador del visualizador, tarea fácil si se utiliza además un CI especializado.

El circuito de control del visualizador puede realizarse mediante circuitos integrados discretos MSI o mediante la utilización de un CI especializado en control de visualizadores, con la ventaja adicional de que éstos suelen admitir varias formas de funcionamiento. El circuito de control debe realizar todas las tareas que sean función del tiempo tales como la gestión del refresco de la pantalla y el control de la transferencia de información entre la memoria de texto y el generador de caracteres, debido a que los microprocesadores no son aún lo suficientemente rápidos en la ejecución de los programas.

Una ventaja adicional que presenta la utilización de un microprocesador consiste en que se puede dotar de cierto grado de inteligencia al sistema y realizar tareas tales como la edición de textos, el control de unidades de almacenamiento

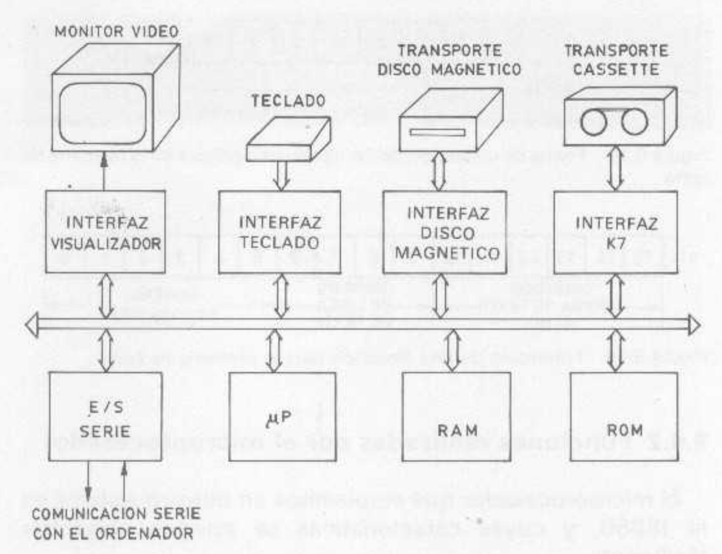


Figura 9.13 Diagrama de bloques para una posible configuración de un terminal inteligente controlado mediante un microprocesador

de información, etc., y si todo este ente constituye un terminal inteligente de otro ordenador, se puede descargar a éste de ciertas tareas elementales (fig. 9.13).

En resumen, en el diseño de un terminal visualizador alfanumérico o gráfico es aconsejable el uso de un microprocesador de ocho bits junto con un CI especializado en el control de visualizadores debido a dos razones principales:

1) Por la flexibilidad y la inteligencia que pueda tener el sistema.
2) Por la facilidad de diseño, y el bajo coste del sistema.

9.4 ESTUDIO DE UN VISUALIZADOR ALFANUMERICO REALIZADO MEDIANTE CI DISCRETOS

En este apartado vamos a realizar el estudio de un sistema visualizador alfanumérico real realizado mediante un microprocesador de ocho bits y circuitos integrados discretos.

Las características generales del visualizador son:

Formato de la pantalla (no programable):

— 16 líneas de caracteres
— 64 caracteres por línea
— Matriz de caracteres de 5×7 puntos.
— Una línea de caracteres está formada en total por 16 barridos del haz electrónico

Memoria de texto:

— Accesible por el microprocesador
— Página fija
— Capacidad: 1024 bytes

Refresco de la pantalla:

— Gestión mediante circuitos lógicos
— Transferencia de información mediante DMA
— Se realiza un refresco cada 20 ms (50 Hz)

Generación del texto:

La generación de una línea de texto se realiza mediante lecturas sucesivas a la memoria de texto (8 por línea)

La generación de la página de texto se realiza mediante lectura secuencial de líneas de texto

— Sincronismo del texto mediante cursor

— Funciones del microprocesador:
  Gestión de la memoria de texto
  Tareas asociadas a los caracteres de control

Veremos a continuación cómo funciona el visualizador, y luego qué circuitos se han incroporado para que responda a la idea de funcionamiento descrita.

9.4.1 Principio general de funcionamiento

Como paso previo en el diseño de cualquier sistema es importante especificar concretamente qué funciones debe realizar, y a poder ser, quién debe realizarlas y cómo.

Dado que en nuestro sistema tenemos dos bloques fundamentales, el microprocesador y el circuito de control, asignamos a cada uno de ellos las siguientes funciones:

Circuitos de control:

— Generación de los caracteres
— Generación de las líneas de texto
— Generación de la página de texto
— Sincronismo de la pantalla basándose en el cursor
— Gestión del refresco de la pantalla

Microprocesador:

— Pretratamiento de los caracteres visualizables
— Gestión del cursor en la memoria de texto
— Realización de las funciones de control asociadas a los siguientes caracteres de control:
  - CR = Retorno de línea
  - LF = Salto de línea
  - NG = Inversión de video
  - PS = Video normal
  - BS = Cursor hacia atrás
  - HT = Tabulación horizontal
  - VT = Tabulación vertical
  - FF = Borrado de todo el texto

Si el formato de la pantalla es de 16 líneas de texto con 64 caracteres en cada línea, la memoria de texto deberá tener un tamaño de 1.024 octetos, almacenando en ella el código ASCII de los caracteres a visualizar. Si como generador de caracteres utilizamos una matriz de 5×7 puntos, tal como la que tiene el CI TMS2501 (fig. 9.14), la longitud mínima de una palabra de la memoria de texto será 6 bits, y el código de los caracteres será el ASCII de 6 bits. En nuestro caso hemos escogido una longitud de palabra de 8 bits, utilizándolos de la siguiente forma (fig. 9.15):

Bit 0 a bit 5: Código ASCII del carácter
Bit 6: Control de fondo
Bit 7: Cursor

El sincronismo de la página visualizada en la pantalla lo realizaremos a partir de la posición relativa del cursor dentro de la memoria de texto si tenemos en cuenta que sólo puede existir un cursor en toda la memoria de texto, tarea de la que se encargará el microprocesador.

Teniendo en cuenta que el microprocesador de ocho bits utilizado direcciona la memoria mediante 16 bits, y basándose en el formato de la pantalla, la relación entre la posición en la pantalla de un carácter y la posición de memoria en que se encuentra su código es la siguiente (fig. 9.16):

Bit 0 a bit 5: n.º de carácter dentro de una línea de texto
Bit 6 a bit 9: n.º de la línea de texto
Bit 10 a bit 15: dirección absoluta de la página de texto

Los registros asociados de que debe disponer el controlador del visualizador serán los siguientes:

— Un registro de 6 bits para el n.º del carácter

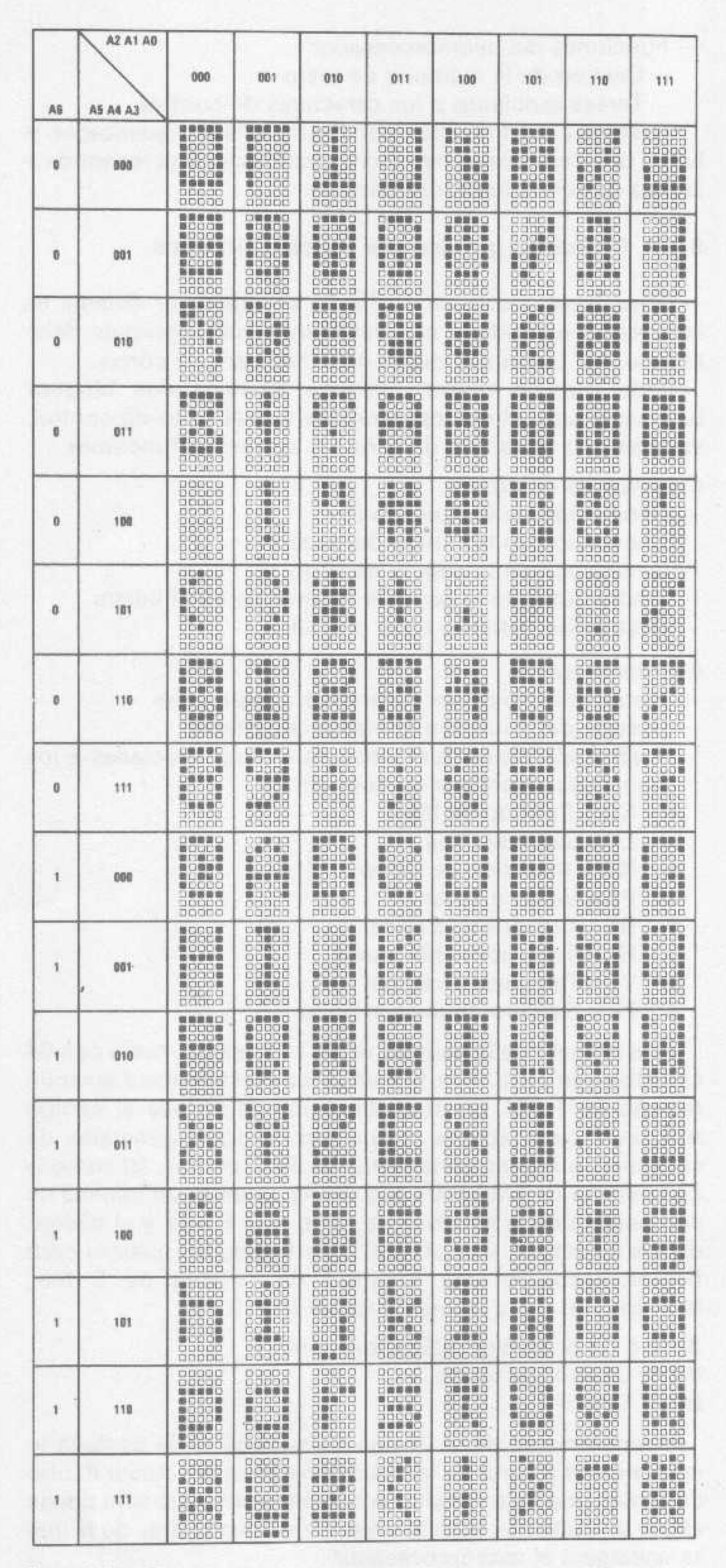


Figura 9.14 Ejemplo de un conjunto completo de caracteres alfanuméricos

— Un registro de 4 bits para el n.º de línea
— Un registro de 4 bits para el n.º de barrido del haz electrónico dentro de una línea de texto

La dirección absoluta de la página de texto está cableada en el circuito controlador, aunque de forma muy fácil podría ser programable mediante la utilización de un registro adicional programable de 6 bits.

Finalmente, la gestión del refresco de la pantalla la realizará el circuito de control utilizando un canal de DMA.



Figura 9.15 Forma de utilización de los bits de una palabra de la memoria de texto



Figura 9.16 Formación de una dirección para la memoria de texto

9.4.2 Funciones realizadas por el microprocesador

El microprocesador que empleamos en nuestro sistema es el I8080, y cuyas características se pueden encontrar fácilmente.

El programa de control del visualizador que corresponde a las tareas asignadas al microprocesador y enumeradas en el apartado anterior se ha desarrollado en el lenguaje de alto nivel PL/M y puede verse su listado en la figura 9.17. El programa de control consta de dos partes básicas: inicialización del visualizador y tratamiento de los caracteres, tanto de los visualizables como de los de control. A continuación se realiza un breve análisis de dicho programa.



```
control$video:
do;
   /*
         registros generales utilizados
   */
   declare cur address public;              /*reg. cursor*/
   declare aux address public;              /*reg. borrado*/
   declare ngr byte public;                 /*reg. fondo*/
   /*
         constantes utilizadas
   */
   declare pripag literally '2000h';        /*inicio pagina de texto*/
   declare fipag literally'pripag+1023';    /*fin pagina de texto*/
   declare long$lin literally '64';         /*longitud de una linea de texto*/
   declare fin$lin literally 'long$lin-1';  /*posicion ultimo caracter/lin*/
   declare bs$car literally '7';            /*caracteres de control video*/
   declare ht$car literally '8';
   declare lf$car literally '9';
   declare vt$car literally '10';
   declare ff$car literally '11';
   declare cr$car literally '12';
   declare ng$car literally '13';
   declare ps$car literally '14';
   declare neg literally '40h';             /*video inverso*/
   declare pos literally '0';               /*video normal*/
   declare crs literally '80h';             /*codigo del cursor*/
   declare sp literally '20h';              /*codigo caracter borrado*/
  /*

   RUTINAS DE UTILIZACION GENERAL

  */
                                            /*rutina que elimina el cursor
                                            en la pagina de texto*/
 quitar$cursor:
   procedure;
   declare mem based cur byte;
   mem=mem and not(crs);
   end quitar$cursor;
                                            /*rutina que coloca de nuevo
                                            el cursor en la pagina de texto*/
 poner$cursor:
   procedure;
   declare mem based cur byte;
   mem=mem or crs;
   end poner$cursor;
                                            /*rutina que adelanta el cursor
                                            dentro de una linea de texto*/
 incr$lin$cur:
   procedure;
   cur=cur+1;
   if (low(cur) and fin$lin)=0 then cur=cur-1;
   end incr$lin$cur;
                                            /*rutina que atrasa el cursor
                                            dentro de una linea de texto*/
 decr$lin$cur:
   procedure;
   cur=cur-1;
   if (low(cur) and fin$lin)=fin$lin then cur=cur+1;
   end decr$lin$cur;
                                            /*rutina que adelanta el cursor
                                            en una linea de texto*/
  incr$pag$cur:
   procedure;
   cur=cur+long$lin;
   if cur>fipag then cur=cur-64*16;
```

Figura 9.17 Programa completo para el control del visualizador, escrito en PL/M

```
  end incr$pag$cur;
                                        /*rutina que borra los caracteres
                                        situados entre las posiciones de
                                        memória (aux) y el final de la linea
                                        de texto a la que apunta el cursor*/
borrar$car:
  procedure;
  declare (aux1,aux2) address;
  declare mem based aux1 byte;
  aux1=aux;
  aux2=cur or fin$lin;
  do while aux1<=aux2;
          mem=sp;
          aux1=aux1+1;
          end;
  aux=aux1;
  end borrar$car;
                                        /*rutina que realiza el salto de
                                        linea de texto*/
saltar$linea:
  procedure;
  call incr$pag$cur;
  call borrar$car;
  end saltar$linea;
/*

  RUTINA DE GESTION DE VISUALIZACION

*/
visualizar:
  procedure (car) public;
  declare (car,i) byte;                 /*'car' es el caracter que le llega
                                        al sistema para ser visualizado

                                        mascaras utilizadas:
                                                              */

  declare mask$ascii$7 literally '01111111b';
  declare mask$ascii$6 literally '00111111b';
  declare mask$asc$min literally '00011111b';
                                   /*                         */
  declare mem based cur byte;           /*'mem' es la posicion de la memoria
                                        de texto donde se pondra el codigo
                                        del caracter a visualizar*/
  car=car and mask$ascii$7;             /*borrar bit de paridad*/
                                        /*caso de caracteres no visualizable
                                        ni de control*/
  if car<bs$car then return;
  if ((car>ps$car) or (car<20h)) then return;
                                        /*caracter visualizable o
                                        caracter de control*/
  call quitar$cursor;
  if car>=20h then do;                  /*caracter visualizable*/
          if car>=60h then car=car and mask$asc$min;
          car=car and mask$ascii$6;     /*paso a codigo ascii de 6 bits*/
          car=car or ngr;               /*se pone control de fondo*/
          mem=car;                      /*se situa el caracter en la memoria
                                        de texto*/
          call incr$lin$cur;            /*se actualiza el cursor*/
          call poner$cursor;            /*se coloca de nuevo el cursor*/
          return;
          end;
                                        /*caracteres de contorl*/
  do case (car-bs$car);
                                        /*retroceder cursor*/
  bs:do;

         call decr$lin$cur;
         call poner$cursor;
         return;
         end;
                                          /*tabulacion horizontal*/
  ht:do;
         call incr$lin$cur;
         do while (cur mod 8)<>0;
                 call incr$lin$cur;
                 if (low(cur)and fin$lin)<>fin$lin then call incr$lin$cur;
                 else goto ht1;
                 end;
         ht1:
         call poner$cursor;
         return;
         end;
                                          /*salto de linea*/
  lf:do;
         call saltar$linea;
         call poner$cursor;
         return;
         end;
                                          /*tabulacion vertical*/
  vt:do;
         do i=0 to 3;
                 call saltar$linea;
                 end;
         call poner$cursor;
         return;
         end;
                                          /*borrado de todo el texto*/
  ff:do;
         do i=0 to 15;
                 call saltar$linea;
                 end;
         call poner$cursor;
         return;
         end;
                                          /*retorno cursor*/
  cr:do;
         do while (low(cur) and fin$lin)<>0;
                 cur=cur-1;
                 end;
         call poner$cursor;
         return;
         end;
                                          /*inversion video*/
```

Figura 9.17 (Continuación)

```
   ns:do;
           ngr=neg;
           call poner$cursor;
           return;
           end;
                                                     /*restablecer video*/
   ps:do;
           ngr=pos;
           call poner$cursor;
           return;
           end;
   end;
   end visualizar;
 /*

   RUTINA DE INICIALIZACION

 */
 inicializar:

         procedure public;
         if ((cur<pripag) or (cur>fipag)) then do;
                 cur=pripag+long$lin;
                 aux=pripag;
                 call visualizar(ff$car);
                 return;
                 end;
         else return;
         end inicializar;
  end controlvideo;
  *
```

Figura 9.17 (Continuación)

9.4.2.1 *Inicialización. Registros utilizados* (fig. 9.18)

La gestión del visualizador requiere solamente tres registros:

- CUR: Contiene la dirección de la memoria de texto en que se encuentra el cursor.
- NGR: Indica si la representación de los caracteres debe hacerse de forma normal o con video inverso.
- AUX: Registro auxiliar utilizado para el borrado de caracteres en la memoria de texto.

y las siguientes constantes:

- PRIPAG: Dirección inicial de la página de texto.
- FIPAG: Dirección final de la página de texto.
- LONGLIN: Longitud de una línea de texto.
- FINLIN: Máscara utilizada para detectar el fin de una línea de texto.
- NEG: Código para representación inversa.
- POS: Código para representación normal.
- CRS: Código del cursor en la memoria de texto.
- SP: Código de un espacio en la pantalla, utilizado en el borrado del texto.

La inicialización del visualizador consiste en borrar toda la memoria de texto escribiendo en ella el código del espacio

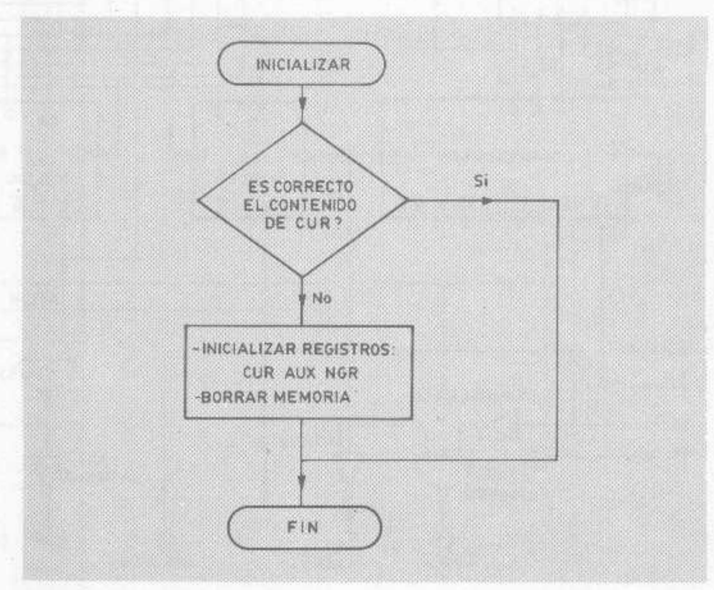


Figura 9.18 Diagrama de flujo para la inicialización del visualizador

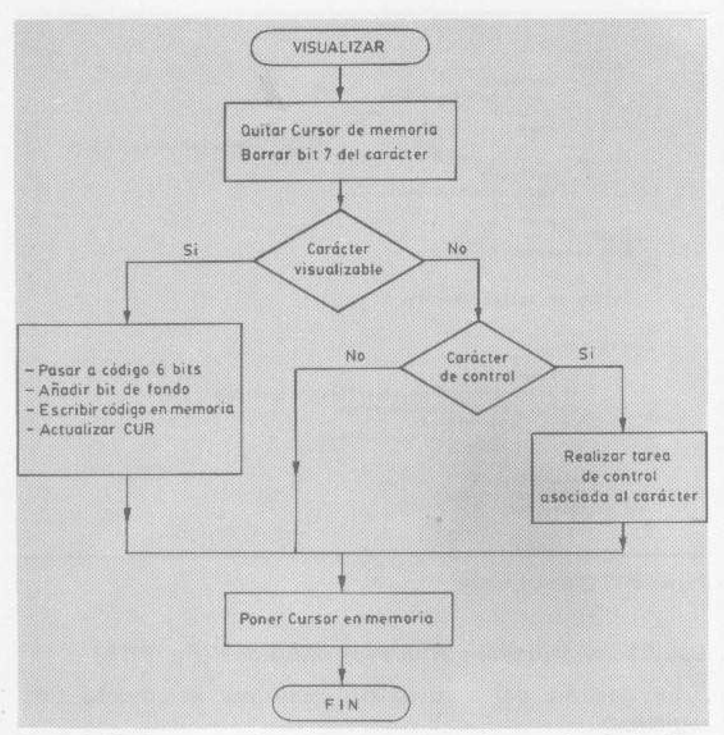


Figura 9.19 Diagrama de flujo para el tratamiento de un carácter a visualizar

SP, y en cargar los tres registros utilizados con unos valores iniciales tal como puede verse en el listado del programa ('procedure inicializar').

La inicialización del visualizador es conveniente que se ejecute de forma automática al poner en marcha el sistema para que no aparezcan caracteres aleatorios en la pantalla. El único inconveniente es que, si luego hacemos un 'reset' del sistema, podría borrarse la pantalla si no tuviésemos en cuenta el estado anterior del visualizador, cosa que se consigue analizando el contenido del registro CUR.

9.4.2.2 *Tratamiento de los caracteres* (fig. 9.19)

Cuando al microprocesador le llega un carácter que en principio se deba visualizar, el microprocesador debe analizarlo e incluirlo dentro de uno de los tres grupos siguientes:

a) Carácter de control
b) Carácter visualizable
c) Carácter no visualizable

Si el carácter es de control, el microprocesador debe ejecutar la tarea asociada al mismo, gestionando la memoria de texto o los registros en la forma adecuada.

Si el carácter es visualizable, el microprocesador debe escribir su código ASCII de 6 bits junto con el bit de fondo en la posición de la memoria de texto que ocupaba el cursor, haciendo avanzar a éste una posición tanto en la memoria de texto como en su registro asociado. El salto de línea no se realiza automáticamente y por tanto en el caso de querer incluir en una línea de texto más caracteres de los que caben físicamente (64), la operación no se realiza y además se bloquea la posición del cursor.

Si el carácter ni es de control ni es visualizable, se desprecia.

Figura 9.20 Esquema del generador de sincronismos y de control

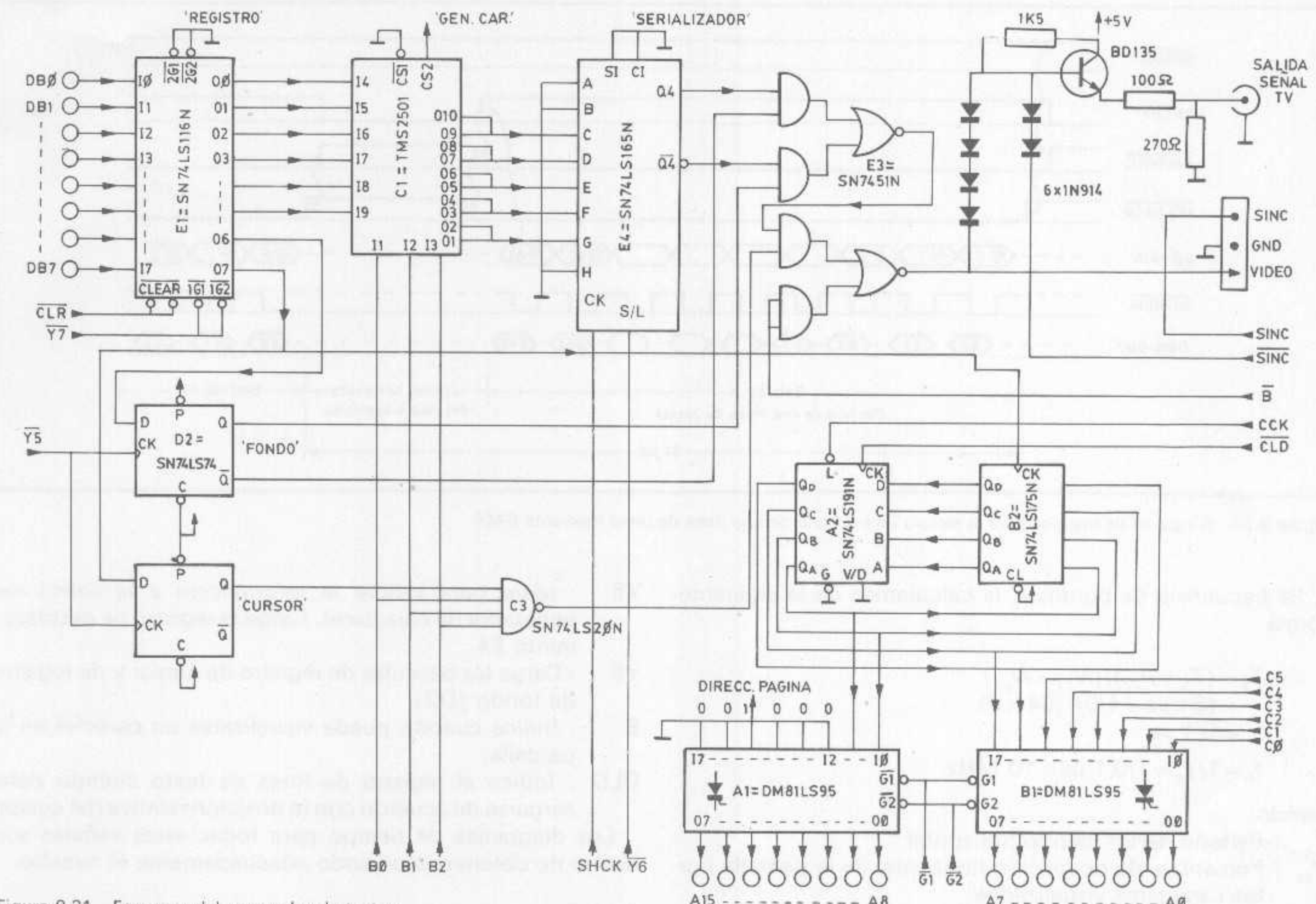


Figura 9.21 Esquema del generador de textos

9.4.3 Funcionamiento del circuito controlador del visualizador

El esquema general del circuito controlador del visualizador puede verse en las figuras 9.20 y 9.21, a las que nos referiremos en lo sucesivo.

A		B	
+5 V	01, 02	+5 V	ALIMENTACIONES
Libres	03, 04	Libres	
+12 V	05, 06	+12 V	
−5 V	07	−5 V	
−12 V	08, 09	−12 V	
Libre	10	Libre	DIRECCIONES
A0	11	A8	
A1	12	A9	
A2	13	A10	
A3	14	A11	
A4	15	A12	
A5	16	A13	
A6	17	A14	
A7	18	A15	
Libre	19	Libre	DATOS CONTROL
$\overline{\text{NODMA}}$	21	Libre	
$\overline{\text{INTREQ}}$	22	$\overline{\text{INTACK}}$	
$\overline{\text{HOLREQ}}$	23	$\overline{\text{HOLDACK}}$	
$\overline{\text{WRITE}}$	24	$\overline{\text{ADMEM}}$	
$\overline{\text{NOTYET}}$	25	$\overline{\text{ADPER}}$	
DB0	26	DB4	
DB1	27	DB5	
DB2	28	DB6	
DB3	29	DB7	
0 V	30, 31	0 V	

Figura 9.22 Señales del bus del sistema microcomputador utilizado

9.4.3.1 *Señales de interconexión con el μP*

El bus del sistema utilizado puede verse en la figura 9.22. Las señales que utiliza el visualizador son las siguientes:

A0-A15: Líneas de direcciones, a través de las que se direcciona la memoria.

DB0-DB7: Líneas de datos, a través de las cuales se transfiere la información en paralelo.

ADMEM: Señal que indica acceso a memoria.

HOLREQ: Señal de petición del bus al microprocesador.

HOLDACK: Señal de concesión del bus por parte del microprocesador.

NODMA: Señal que impide la utilización del bus por parte de cualquier periférico.

Suponemos que solamente el controlador del visualizador es el único periférico que utiliza acceso directo a memoria, por lo que no se mencionan las señales de arbitraje del bus.

El diagrama de tiempos para la adquisición del bus por parte del controlador del visualizador y la posterior lectura de los códigos de una línea de caracteres puede verse en la figura 9.23.

9.4.3.2 *El generador de sincronismos. Registros y contadores*

La base de tiempos del circuito controlador reside en un oscilador de cristal de cuarzo que trabaja a 20 MHz y a cuya salida hay un divisor de frecuencia por 2 que nos da una señal cuadrada *SHCK* de 10 MHz y que corresponde a la frecuencia de puntos f_p (fig. 9.20).

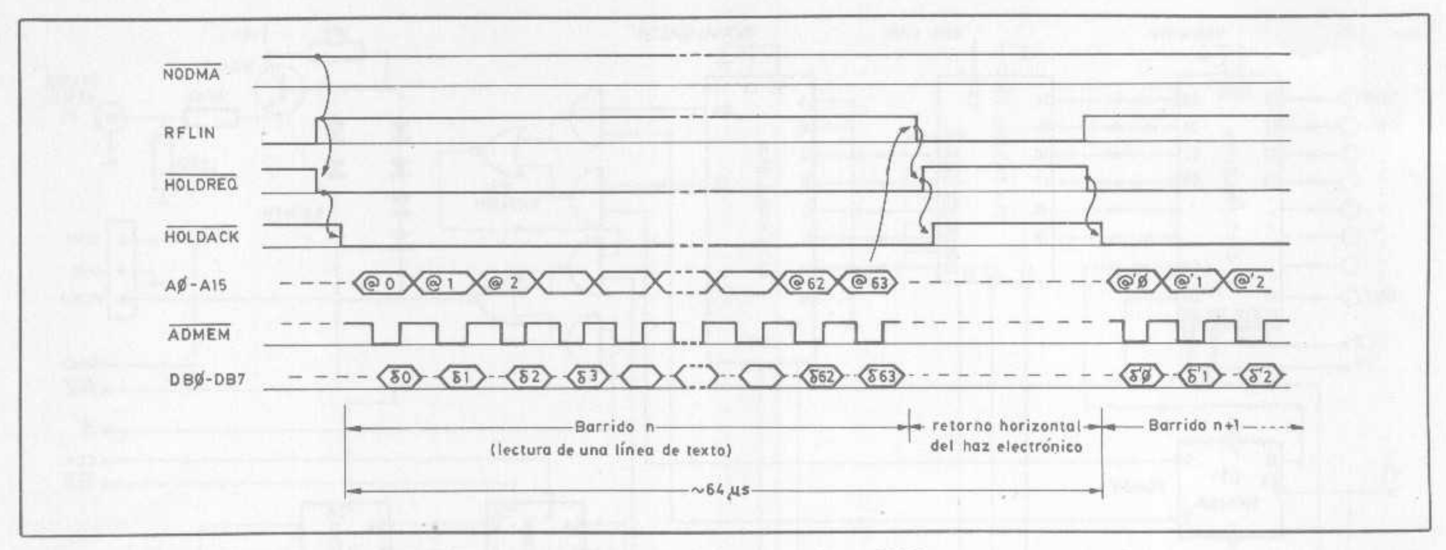


Figura 9.23 Diagrama de tiempos para la lectura en memoria de una línea de texto mediante DMA

La frecuencia de puntos f_p la calculamos de la siguiente forma:

$$T_p = (T_1 \times P_{oc}) / (N_{c1} \times N_{pc})$$
$$= (64\ \mu s \times 4/5) / (64 \times 8)$$
$$= 0{,}1\ \mu s$$
$$f_p = 1/T_p = 1/0{,}1\ \mu s = 10\ MHz$$

siendo:

T_1 : Período de un barrido horizontal

P_{oc} : Porcentaje de ocupación horizontal de la pantalla por los caracteres visualizados

N_{c1} : Número de caracteres por línea

N_{pc} : Número de puntos totales horizontales para un carácter visualizado (5 del carácter y 3 de separación entre caracteres).

La señal *SHCK* será la señal de reloj para el registro de desplazamiento E4 situado a la salida del generador de caracteres C1.

Si dividimos *SHCK* por 8 obtenemos la frecuencia de un carácter (señal *D2*) que nos controla un contador de 4 bits (1/2 A3) en cuya salida tenemos la dirección relativa dentro de una línea de texto del carácter visualizado (señales *C0* a *C3*).

Las señales de frecuencia de línea (15.625 Hz) y de cuadro (50 Hz) las obtenemos aproximadamente utilizando los contadores/divisores A3, A4, B4, C2 y D4, obteniendo al final la señal de sincronismo compuesto SINC.

Los registros más importantes asociados al generador de sincronismos son los siguientes:

Registro del número del cárácter. Indica el número del carácter dentro de una línea de texto. Lo constituye la mitad del circuito A3 y las señales son *C0* a *C3*.

Registro del número de fila visualizada. Indica qué fila de la matriz de puntos está siendo visualizada. Lo constituye la otra mitad del circuito A3 y las señales son *B0* a *B3*. Si *B3* = 1 lógico, indica que el barrido corresponde a uno de separación entre líneas de texto.

Registro del número de línea de texto. Indica qué línea de texto está siendo visualizada. Lo constituye el circuito A2 y B2, y las señales son las cuatro salidas Q_i de A2.

Del generador de sincronismos se obtienen asimismo las señales de control para el generador de textos:

Y7 : Indica cuándo los datos son válidos en el bus y pueden almacenarse en el registro E1.

Y6 : Idem. para validar la información a la salida del generador de caracteres. Carga el registro de desplazaminto E4.

Y5 : Carga las básculas de registro de cursor y de registro de fondo (D2).

B : Indica cuándo puede visualizarse un carácter en la pantalla.

CLD : Indica al registro de línea de texto cuándo debe cargarse de acuerdo con la posición relativa del cursor.

Los diagramas de tiempo para todas estas señales son fáciles de obtener estudiando adecuadamente el circuito.

9.4.3.3 *El generador de texto* (fig. 9.21)

Poco antes de iniciar la visualización de una línea de texto, el generador de sincronismos pide el control del bus al microprocesador, y una vez que lo obtiene se van generando las direcciones sucesivas de la memoria de texto correspondientes a los caracteres a visualizar según el estado de los registros y del amplificador de bus A1 y B1, obteniéndose los códigos de los caracteres a través de las líneas de bus DB0 a DB7, hasta un total de 64, momento en que se deja libre el bus.

El código del carácter a visualizar se guarda temporalmente en el registro del ocho bits E1 junto con la información del fondo y del cursor. La salida de este registro está conectada a la entrada del generador de caracteres C1, controlado mediante las señales B0, B1 y B2 que le indican el número de fila de la matriz que debe visualizarse. La información de la fila se guarda en el registro de desplazamiento de ocho bits E4, el cual tiene 3 de sus 8 líneas de entrada de información a nivel 0 y corresponden a los puntos de separación entre dos caracteres consecutivos.

El registro de desplazamiento E4 se carga mediante la señal Y6 y los pulsos de desplazamiento se los proporciona la señal *SHCK* cuya frecuencia es la frecuencia de puntos. A la salida tenemos ya la información serie a partir de la cual se genera la señal de video en función del bit de fondo asociado al carácter y que se ha almacenado temporalmente en una báscula, y de la señal de validación de video *B*.

Cuando se detecta la presencia de cursor en un carácter se almacena el número de línea de texto en que se encuentra el el registro de 4 bits B2, transfiriéndose su contenido al contador/registro programable A2 de número de línea de texto visualizada al terminar la visualización de la página de

texto. La dirección de la primera línea de texto a visualizar en el siguiente refresco de la pantalla es la siguiente a la almacenada en B2, consiguiéndose de esta forma sincronizar la imagen.

La página de texto en nuestro sistema es fija ya que su dirección está cableada (ver entrada de A1), pero podría ser variable utilizando un registro adicional de 6 bits programable por el microprocesador.

Finalmente, puede observarse que a la salida se obtiene la señal de video compuesta por la señal de sincronismo SINC más la señal de salida del registro de desplazamiento E4, tratada adecuadamente para que pueda atacar a un monitor de video o al amplificador de video de un receptor comercial de TV. Este circuito controlador no lleva ningún modulador de HF y por tanto su salida no puede conectarse a la entrada de antena de un receptor de TV.

9.5 UTILIZACION DE CI ESPECIALIZADOS EN CONTROL DE VISUALIZADORES TRC

Los avances que se han producido últimamente dentro del campo de la Microelectrónica traducidos en mayores niveles de integración, junto con el éxito alcanzado por los visualizadores TRC alfanuméricos y gráficos, han hecho posible la aparición de diversos circuitos integrados LSI especializados en el control de visualizadores.

La utilización de dichos CI implica la reducción aproximada en un factor de 3 del número de componentes empleados en el circuito controlador, junto con otras ventajas adicionales tales como la facilidad de diseño, la reducción del coste, etc. (tabla 9.2).

No todos los CI controladores presentan exactamente las mismas características ni tienen integradas las mismas funciones, sino que más bien cada fabricante ha optado por una solución diferente y que suele adaptarse a la configuración de una familia de microprocesadores. Por tanto, antes de elegir qué CI se utilizará en un diseño concreto, deben sopesarse todos los aspectos que puedan influir en la elección.

En este apartado lo que se intentará será dar una visión global de aspectos tales como los factores que pueden influir en la elección del CI, características generales de los CI y soluciones prácticas de diseño presentadas por los fabricantes. No se presenta ningún diseño concreto debido a que cada fabricante suele proporcionar el circuito de aplicación de su CI y normalmente basta con introducir pequeñas modificaciones para que se adapte a un sistema concreto.

9.5.1 Características generales de los CI

Uno de los primeros CI aparecidos, especializado en control de visualizadores, el SFF96364, que tiene incorporadas solamente las funciones de generador de sincronismos y señales de control, con los respectivos registros. La idea de creación de este CI reside en integrar en una sola pastilla todas aquellas funciones que requieren una mayor cantidad de circuitos lógicos discretos para su realización, sin tratar de obtener prestaciones adicionles a las de un visualizador TRC básico.

Este CI no es programable, utiliza memoria local, y difícilmente puede usarse conjuntamente con un microprocesador, por lo que es indicado para usarlo como controlador de un terminal visualizador elemental y totalmente pasivo.

El resto de los CI aparecidos no presentan ya la rigidez de utilización del CI anterior y normalmente están pensados para poder ser utilizados conjuntamente con un microprocesador (figura 9.24).

No obstante, no llevan integradas todas las funciones básicas que debe realizar un circuito controlador de un visualizador, teniéndose que suplir con circuitos lógicos adicionales.

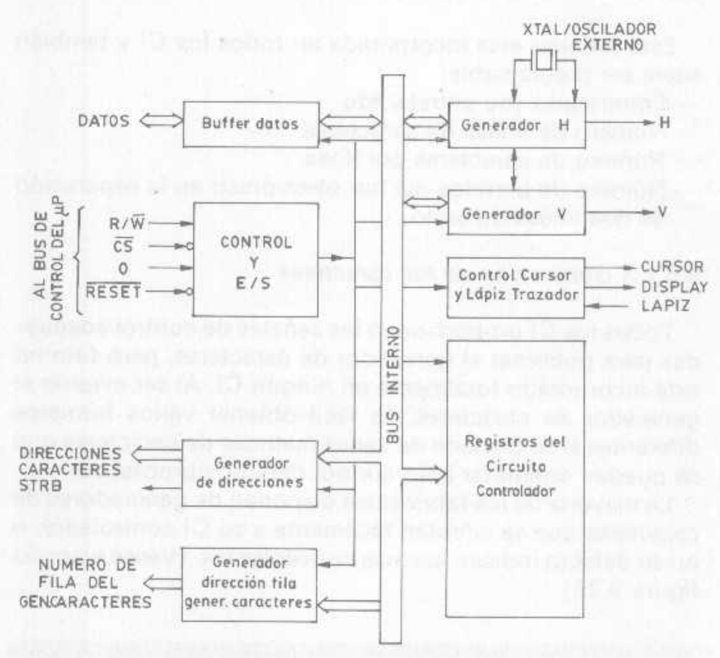


Figura 9.24 Diagrama general de bloques de un CI especializado en control de visualizadores TRC

	Control directo del microprocesador	*Hardware especializado*	*Hardware integrado (CI controladores)*
Utilización	Kits sencillos Gestión pobre	Sistemas medios Gestión media	Cualquier sistema Gestión en función del CI
Ventajas	Sencillez Pocos componentes	Poca ocupación del microprocesador (menos del 33 %) Pretratamiento soft	— Idem. anterior Mantenimiento fácil Funciones programables Fiabilidad Interfaz directo en algunos casos
Inconvenientes	Toda la gestión la debe realizar el µP y no puede utilizarse para otra cosa	Diseño caro y difícil Gestión caracteres control por soft.	No son universales — Precisan de otros CI adicionales

Tabla 9.2 Tabla comparativa de prestaciones para diversas formas de control de un visualizador TRC

La mayoría de los CI están además diseñados de tal forma que se adaptan fácilmente al bus de una familia concreta de microprocesadores, presentando las mismas señales de control, aunque mediante una pequeña transformación de éstas pueden acoplarse a cualquier otro microprocesador.

A continuación pasamos a realizar un breve análisis de cómo están realizadas las funciones de control en los CI aparecidos.

9.5.1.1 *Generación de sincronismos*

Todos los CI llevan incorporada esta función y normalmente proporcionan las señales *H* y *V* y la compuesta de ambas.

Una función adicional que suelen presentar es la de tener programables los tiempos de las señales de sincronismo tales como los siguientes:

— Frecuencia de cuadro: 50-60 Hz
— Ancho de pulso de sincronismo de la señal *H*.
— Idem. de la señal *V*
— Retardo en el inicio de una página de texto
— Idem. en una línea de texto

9.5.1.2 *Formateado de la pantalla del visualizador*

Esta función está incorporada en todos los CI y también suele ser programable:

— Entrelazado (no entrelazado
— Número de líneas de caracteres
— Número de caracteres por línea
— Número de barridos del haz electrónico en la separación de dos líneas de texto.

9.5.1.3 *Generación de los caracteres*

Todos los CI proporcionan las señales de control adecuadas para gobernar al generador de caracteres, pero éste no está incorporado totalmente en ningún CI. Al ser externo el generador de caracteres, es fácil obtener varios formatos diferentes si se dispone de varias matrices de caracteres que se pueden conmutar bajo control del microprocesador.

La mayoría de los fabricantes disponen de generadores de caracteres que se adaptan fácilmente a su CI controlador, o en su defecto indican los más convenientes. (Véase ejemplo figura 9.25).

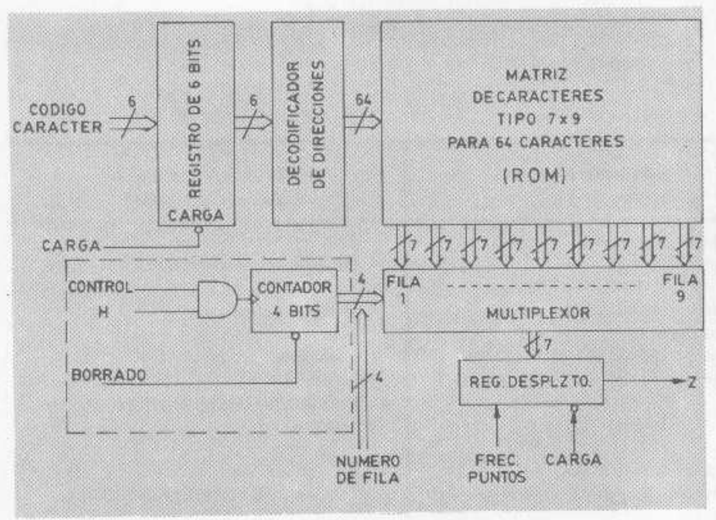


Figura 9.25 Diagrama general de bloques de un CI generador de caracteres que incorpora el serializador

9.5.1.4 *Posibilidad gráfica*

Si una de las matrices de caracteres es del tipo gráfico, el visualizador puede convertirse asimismo en gráfico o semigráfico, ya que el tratamiento de las señales para un visualizador gráfico es similar al de los visualizadores alfanuméricos.

9.5.1.5 *Generación de líneas de texto*

La generación de una línea de texto se realiza mediante lecturas sucesivas a la memoria de texto tantas veces como barridos del haz electrónico corresponden a una línea de texto.

No obstante, algunos proporcionan las señales de control para un buffer circular de línea de texto externo. Si el acceso a la memoria de texto es transparente, no se gana nada poniendo un buffer circular; si el acceso es mediante DMA, se reduce el tiempo de utilización del bus por lo menos en un factor de 7 usando el buffer circular.

9.5.1.6 *Generación de páginas de texto*

La generación de páginas de texto admite todas las formas ya indicadas:

— Lectura secuencial de la página
— Lectura aleatoria de líneas de texto, programable por el microprocesador
— Conmutación de página de texto

Los CI pueden direccionar varias páginas de texto, y lo que se hace es mandar antes de cada refresco de la pantalla la dirección inicial de la página de texto a visualizar desde el microprocesador al CI. La sincronización entre ambos suele hacerse mediante interrupciones.

9.5.1.7 *Cursor*

Normalmente todos los CI llevan incorporadas las funciones del cursor y éste es móvil dentro de la página de texto visualizada modificando adecuadamente el registro del cursor mediante ciertos caracteres de control.

El registro del cursor puede ser escrito y leído por el microprocesador, con lo que resulta fácil saber al microprocesador dónde debe introducir los nuevos caracteres a visualizar.

9.5.1.8 *Lápiz luminoso trazador*

Mediante el uso de un detector luminoso externo conectado al CI controlador es posible identificar caracteres visualizados sobre la pantalla, o trazar dibujos si se utiliza el visualizador en la forma gráfica. Todo es cuestión de utilizar los programas adecuados.

9.5.1.9 *Refresco de la pantalla y utilización de la memoria de texto*

La memoria de texto suele formar parte de la accesible por el microprocesador y el CI controlador puede acceder a ella de dos formas:

a) Mediante DMA, aunque el controlador de DMA no suele estar incorporado al CI controlador.
b) Mediante acceso transparente sincronizado con el microprocesador. El árbitro necesario no suele estar integrado en el CI controlador.

Hay algunos CI controladores que admiten que la

Modelo	Fabricante	μP bus compatible	Reloj	Formato pantalla /programable	Sincronismos programables	Gráfica	Lapiz trazador	Encaps. aliment.	Generador caracteres compatibles	Acc. memoria (modo)	Acc. memoria (forma)	Entrelazado	Comentarios
5027 VTAC	S.M.C.	General	4 MHz	204 c/l /Sí	Sí	No	Sí	NMOS 40p +5V +12V	7002 8002	Transp.	Secuencia Aleatorio	Sí	Posibilidad de funcionamiento autónomo
9927	T.I.	General	4 MHz	204 c/l /Sí	Sí	No	Sí	NMOS 40p +5 V +12V	7002 8002	Transp.	Secuencia Aleatorio	Sí	Compatible 5027 de S.M.C.
5037	S.M.C.	General	4 MHz	204 c/l Sí	Sí	No	Sí	NMOS 40p +5V +12V	7002	Transp.	Secuencia	Sí	Mismo que 5027, pero con entrelazado balanceado
9937	T.I.	General	4 MHz	204 c/l /Sí	Sí	No	Sí	NMOS 40p +5V +12V	7002	Transp.	Secuencia	Sí	Compatible 5037 de S.M.C.
5057	S.M.C.	General	4 MHz	204 c/l /Sí	Sí	No	Sí	NMOS 40p +5V +12V	7002	Transp.	Secuencia	Sí	Mismo que 5037 pero con bloqueo de línea
96364	Thomson	General	1,6 MHz	/No	No	No	No	NMOS 28p +5V	General	Control directo	Secuencia Mem. local	No	Automata no programable
96364	S.M.C.	General	1,6 MHz	/No	No	No	No	NMOS 28p +5V	General	Control directo	Secuencia Mem. local	No	Compatible 96364 de Thomson
9006-135	S.M.C.	General	4 MHz	Hasta 135 col. /No		Sí	No	NMOS 24p					Filas de 1-135 caracteres
9006-83	S.M.C.	General	4 MHz	Hasta 83 col. /No		Sí	No	NMOS 24p					Mismo 8006-135, pero de 83 columnas.
9007-VPAC	S.M.C.	General	4 MHz	/Sí	Sí	Sí	Sí	NMOS	General	D.M.A.	Progr.	Sí	Incorporar D.M.A. Modos multiples
8275	Intel	8080/ 8088	2-3 MHz	80 c/l /Sí	Sí	No	Sí	NMOS 40p +5V	General	D.M.A.	Secuencia Aleatorio		Precisa control externo de D.M.A.
8276	Intel	8080/ 8088	2-3 MHz	80 c/l /Sí	Sí	No	No	NMOS 40p +5V	General	D.M.A.			Precisa control externo de D.M.A.
8350	National	8080/ 8088	2 MHz	110 c/l /Sí	No	No	No	IIL 40p +5	DM8676	D.M.A.	Secuencia Aleatorio		D.M.A. incorporado
6845	Motorola	6800	Sí	/Sí	Sí	Sí	Sí	NMOS 40p +5V	General	Transp.	Secuencia Aleatorio	Sí	Alfanumérico / Gráfico
6845S	Hitachi	6800	Sí	/Sí	Sí	Sí	Sí	NMOS 40p +5V	General	Transp.	Secuencia Aleatorio		Mismo que 6845 de Motorola pero con VSYNC programable
6545	Rockwell Synerteck	6800	Sí	/Sí	Sí	Sí	Sí	NMOS 40p +5V	General	Transp.	Secuencia Aleatorio		Mismo que 6845 de Motorola pero con árbitro de memoria
S68045	AMI	6800	Sí	/Sí	Sí	Sí	Sí	NMOS 40p +5V	General	Transp.	Secuencia Aleatorio		Compatible 6845 de Motorola
2672 PVTC	Signetics	General	4 MHz	/Sí	Sí	Sí	Sí	NMOS 40p +5V	2670 DCGG 2673 VAC	D.M.A.	Varios modos	Sí	Incorpora D.M.A.
8277	National	General	3 MHz	110 c/l /Sí	Sí	Sí	Sí	NMOS 48p +5V	General	D.M.A.	Secuenc. Aleatorio	Sí	Incorpora D.M.A.
1861	R.C.A.	1802		Hasta 64 × 128/Sí	Sí			NMOS 24 p +5	General	D.M.A.			Control D.M.A. externo
1869 1870	R.C.A.	1800	3,2 MHz	40 c/l /Sí	Sí	Sí	No	CMOS 40p +5V	General				Para color RGTS, NTSC, PAL
1876	R.C.A.	1800	6,4 MHz	40 c/l /Sí	Sí	Sí	No	CMOS 40p +10V	General				Para color RGTS, NTSC, PAL
8052 CRT	Zilog	Z-Bus	4-6 MHz	/Sí	Sí	No	No	NMOS 40p +5V	General				Fabricado tmbién por AMD
8869	Fujitsu	General	3 MHz	/Sí	Sí	No	Sí	CMOS 64p +5V		D.M.A.			La información del display lo obtiene por D.M.A.

Tabla 9.3 Tabla comparativa de los CI especializados en control de visualizadores TRC

memoria de texto pueda ser local por lo que pueden trabajar sin la ayuda de un microprocesador. La programación inicial del CI se lleva a cabo entonces mediante la ayuda de una memoria ROM externa en la que están grabados los parámetros de funcionamiento. Esta posibilidad de funcionamento se justifica en que así se pueden realizar visualizadores muy sencillos, con la salvedad de que su forma de utilización es entonces totalmente rígida.

9.5.1.10 *Tecnología de fabricación y alimentaciones*

La mayor parte de los CI están realizados en tecnología

NMOS con una tensión de alimentación única de +5 voltios.

Se proporciona una tabla comparativa de características entre los diversos CI controladores en la la tabla 9.3.

9.5.2 Control software del visualizador TRC

Supongamos que estemos en el caso de disponer de un terminal semiinteligente de un ordenador controlado mediante un microprocesador y que incorpore por lo menos un teclado, un visualizador alfanumérico y un canal de comunicación con el exterior.

Si el control del visualizador se realiza mediante un CI especializado, es imprescindible que el microprocesador programe al CI al poner el sistema en marcha, que se encargue de gestionar los terminales, y de dar los parámetros adecuados cuando el CI controlador del visualizador se los pida.

La forma más eficiente de sincronización entre el microprocesador y los periféricos es mediante la utilización de las interrupciones, ya que así se evitan las pérdidas de tiempo en los bucles de espera.

La rutina de inicialización del sistema podría estar estructurada según la figura 9.26 y las funciones básicas a realizar serían las siguientes:

a) Inicialización de los registros del microprocesador.
b) Borrar la memoria de texto del visualizador.
c) Inicializar el CI controlador del visualizador programándole adecuadamente los registros.
d) Inicializar, si se debe, el resto de los periféricos.
e) Inicializar el sistema de interrupciones del microprocesador, quedándose luego a la espera de recibir órdenes de los diversos periféricos.

INICIALIZACION
INICIALIZAR REGISTROS MICROPROCESADOR
BORRAR MEMORIA DE TEXTO
INICIALIZAR CONTROLADOR T R C: - CURSOR - FORMA FUNCTO. - DEMAS REGISTROS
INICIALIZAR OTROS PERIFERICOS
PREPARAR INTERRUPCIONES
INTERRUPCION
SI
NO
② FIN SERVICIO DE INTERRUPCION
① A LA SUBRUTINA DE INTERRUPCION

Figura 9.26 Diagrama de flujo para la rutina de inicialización de un sistema con visualizador TRC controlado mediante CI especializado

Cuando al microprocesador le llega una petición de interrupción estando disponible, primero mira qué periférico es el que la pide (fig. 9.27) y a continuación pasa a prestarle el servicio adecuado.

Si se trata de una interrupción que proviene del teclado o del sistema de comunicaciones con el exterior (fig. 9.28), los tipos de tareas que debe realizar el microprocesador se reducen básicamente a tres:

1) Visualizar en la pantalla un nuevo carácter. En esta tarea el microprocesador escribe el código del carácter a visualizar en la memoria de texto y en la posición donde le indica el registro de cursor, actualizando a continuación el mismo.
2) Realizar una función de control sobre el visualizador. El microprocesador da las órdenes oportunas al CI controlador del visualizador, o modifica la memoria de texto, según proceda.
3) Realizar otras funciones que no conciernen al visualizador y que no tenemos en cuenta.

Las interrupciones que provienen del CI controlador (fig. 9.29) del visualizador suelen ser normalmente para pedir al microprocesador la dirección inicial del texto a visualizar en una nueva imagen o en una nueva línea de texto, de acuerdo con la forma en que se haya programado, y no presentan ninguna complicación especial.

Cuando el microprocesador ha ejecutado la tarea pedida mediante la interrupción, puede pasar de nuevo al estado de espera.

Un punto que es importante tener en cuenta es la prioridad de las interrupciones asignadas a cada periférico en función del tiempo máximo de que dispone el microprocesador en prestarle el servicio adecuado. La prioridad del CI controlador del visualizador debe ser de las más altas ya que en caso contrario la imagen no se refrescaría adecuadamente. No

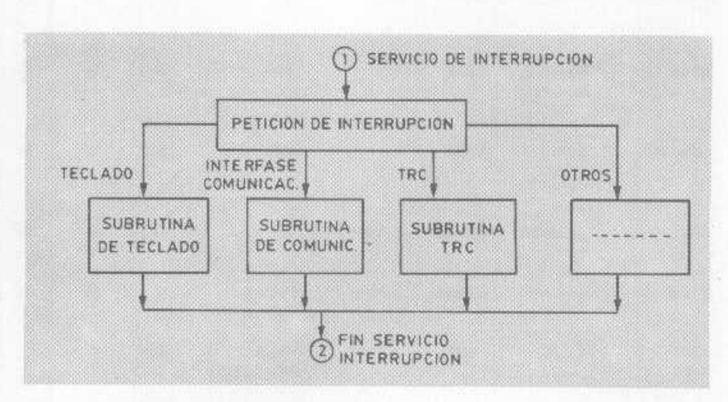


Figura 9.27 Rutina de servicio de interrupción

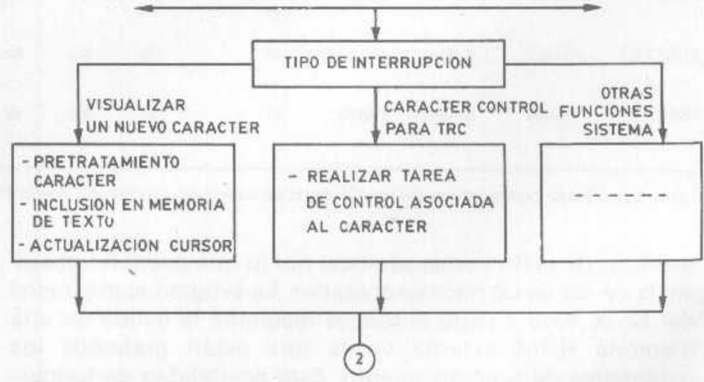


Figura 9.28 Rutinas de servicio de teclado y de comunicación con el exterior

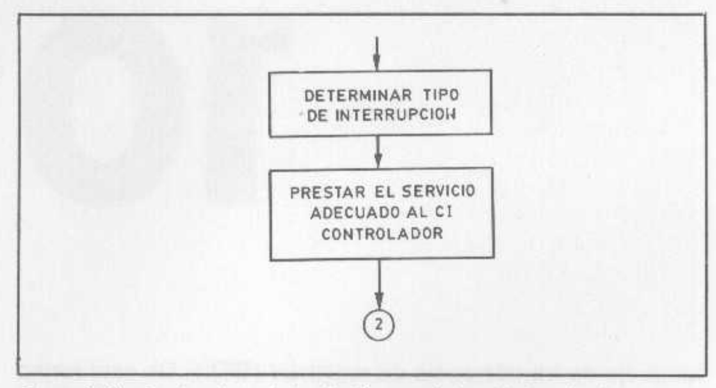


Figura 9.29 Rutina de servicio del CI controlador del visualizador

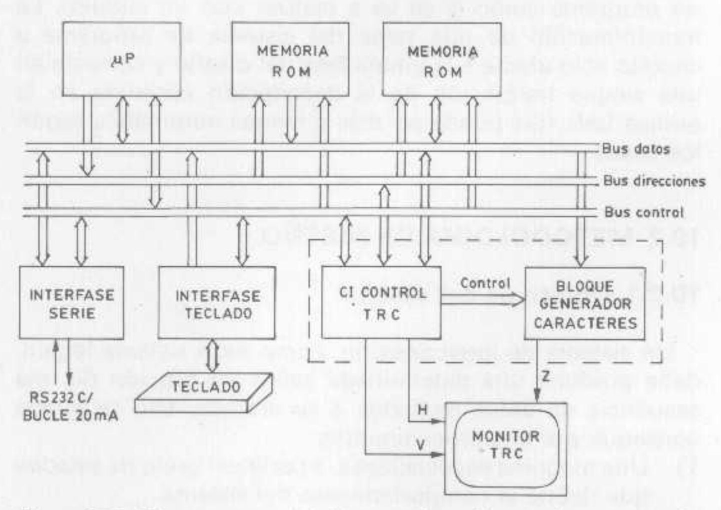


Figura 9.30 Diagrama general de bloques de un sistema microcomputador que utiliza un CI especializado en control de visualizadores TRC

hay ningún criterio exacto para fijar las diversas prioridades y éstas dependen de la configuración global de cada sistema.

9.5.3 Realizaciones prácticas

Debido a que cada fabricante proporciona suficiente información sobre su CI controlador de visualizadores, solamente damos una idea global de cómo funcionan éstos, una lista general de características, y algunos circuitos de aplicación a nivel de diagramas de bloques.

No hay ningún CI controlador de visualizadores que lleve incorporadas todas las funciones y por tanto tampoco existe ninguna prioridad en cuanto a aconsejar un CI u otro. Como únicas ideas generales aplicables en la elección, solamente podemos dar las siguientes:

1) Escoger aquel CI que proporcione una forma de funcionamiento lo más próxima posible a la idea que tengamos preestablecida.
2) Escoger aquel CI que se adapte más fácilmente al bus del microprocesador empleado.
3) En resumen, escoger aquel CI que nos presente menos problemas en el momento del diseño.

Una vez que el diseñador escoja el CI, es aconsejable seguir las indicaciones que el fabricante proporciona en sus notas de aplicación, ya que así se consigue un diseño con una cantidad de componentes mínima.

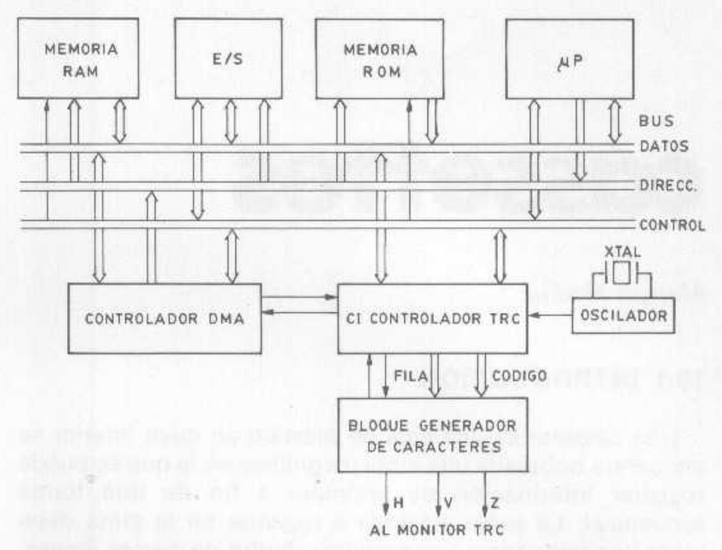


Figura 9.31 Sistema visualizador que utiliza acceso directo a memoria

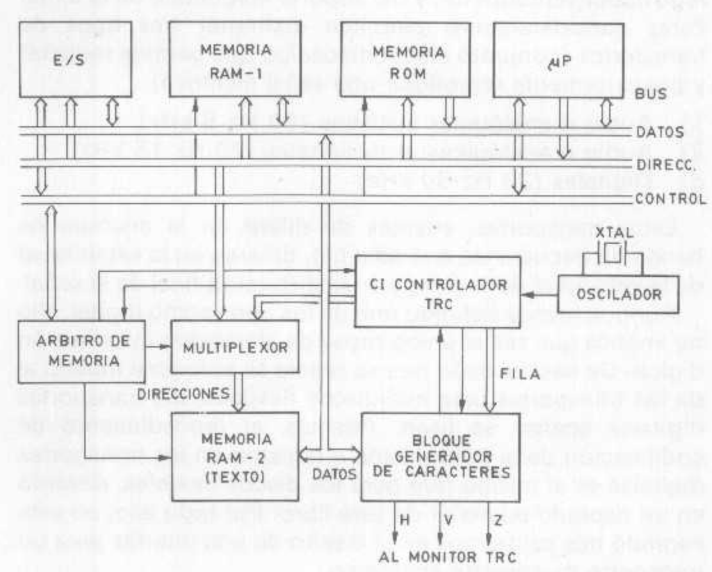


Figura 9.32 Sistema visualizador que utiliza acceso transparente a la memoria de texto

En las figuras 9.30, 9.31 y 9.32 pueden verse algunos ejemplos de circuitos de aplicación de los CI controladores a nivel de diagrama de bloques.

BIBLIOGRAFIA

[1] Mise en œuvre des microperiphériques. Université Pierre et Marie Curie. Inst. Programation París - 1978.
[2] Alphanumeric TV display interface. Micro Scope n.º 6 - Noviembre 1976. pp. 1-6 Swiss Federal Institute of Technologie. Laussanne-1976.
[3] Interfaces de visualisation. Partes I, II, III y IV Minis et Micros n.º 66, 67, 68, 69. París 1978.
[4] Microprocessor interfacing techniques. Lesee y Zaks (Sybex). 1978.
[5] CRT Controllers. EDN - Diciembre 1978.
[6] Proyecto Fin de Carrera. J.M. Pallarés Curto. E.T.S.I. Telecomunicación. Barcelona-1978.
[7] Terminal design with CRT controller IC's. Charles Carinalli - National Semiconductors. 1978.
[8] Notas de aplicación de diversos fabricantes de CI controladores de TRC.
[9] El visualizador TRC. J. Bosch Solsona. Satelec-79. Madrid.

10 cassettes

Manuel Medina

10.1 INTRODUCCION

Una cassette es una caja de plástico en cuyo interior se encuentra bobinada una cinta magnética en la que se puede registrar información de principio a fin de una forma secuencial. La señal eléctrica a registrar en la cinta debe tener una frecuencia comprendida dentro de ciertos límites, que dependen de la calidad de los amplificadores del sistema registrador/reproductor, y del soporte magnético de la cinta. Estas consideraciones permiten distinguir tres tipos de transportes (conjunto electromecánico que permite registrar y posteriormente reproducir una señal eléctrica):

1) Audio o analógicos sencillos (50 Hz-5 kHz)
2) Audio o analógicos profesionales (20 Hz-15 kHz)
3) Digitales (20 Hz-30 kHz)

Estos transportes, además de diferir en la anchura de banda de frecuencias que admiten, difieren en la estabilidad de la velocidad de la cinta y en la distorsión final de la señal.

Aunque hemos definido uno de los tipos como digital, ello no implica que sea el único capaz de almacenar información digital. De hecho, dado que su precio se aproxima mucho al de los transportes para minidiscos flexibles, los transportes digitales apenas se usan. Además, el procedimiento de codificación de la información a registrar en los transportes digitales es el mismo que para los discos flexibles, descrito en un capítulo posterior de este libro. Por todo ello, en este capítulo nos centramos en el diseño de una interfaz para un transporte de cassette analógico.

Dado el reducido ancho de banda y la poca estabilidad del arrastre de la cinta en los transportes analógicos, no se puede pretender registrar en ellos la información en forma digital, mediante modulación de impulsos, como sucede en los transportes digitales (cintas o discos magnéticos). En los transportes analógicos la señal binaria a registrar se debe codificar empleando modulación de amplitud o de frecuencia. Problemas de ajuste de volumen han hecho descartar los sistemas modulados en amplitud y todos los procedimientos empleados, «normalizados» a nivel internacional, emplean modulación de frecuencia. En el apartado 10.4 (Formatos de grabación) se describen los procedimientos más frecuentemente empleados.

Este capítulo se divide en dos partes. En la primera se hace una descripción general del método de diseño empleado. En a segunda se describe la realización del sistema de interconexión en tres versiones que van desde un sistema con el mínimo de circuitería y máximo programa, hasta un sistema realizado casi completamente con circuitos y un programa mínimo. El método de diseño descrito está orientado tanto a la realización de la parte programada (software) como de la parte física (hardware) del sistema. Básicamente consiste en describir detalladamente las operaciones a realizar por el sistema, empleando un pseudolenguaje fácilmente traducible a ensamblador y/o a lenguaje de transferencia de registros (RTL). De esta forma la primera fase del diseño, en la que se concreta el algoritmo que debe ejecutar el sistema, sirve tanto si se va a realizar con un programa como si se va a realizar con un circuito. La transformación de una parte del sistema de programa a circuito sólo afecta a la última fase del diseño y consiste en una simple traducción de la descripción obtenida en la primea fase, que puede ser más o menos automática según los casos.

10.2 METODOLOGIA DE DISEÑO

10.2.1 Objetivos del diseño

Un sistema de interconexión, como todo sistema lógico, debe producir una determinada salida en función de una secuencia de datos recibidos a su entrada. Ello se puede conseguir por tres procedimintos:

1) Una máquina especializada, a partir del grafo de estados que define el comportamiento del sistema.
2) Una máquina programable, escribiendo el programa que maneje las entradas y salidas, a partir de una descripción del algoritmo que realiza el sistema.
3) Una máquina microprogramada. Esta solución puede llegar a confundirse con la primera si se diseña la máquina «a medida», o con la segunda si se emplea un procesador microprogramable.

10.2.2 Descripción de la solución

Tradicionalmente se emplea un método de diseño distinto para cada procedimінto, pero vamos a ver que se puede emplear un solo método descendente, que tan solo divergirá en la fase final de la realización.

Este método de diseño común consiste en describir estructuradamente el algoritmo que debe realizar el sistema, descendiendo en la descripción hasta el nivel de transferencia de datos entre registros y puertos de entrada/salida.

Esta descripción puede ser suficiente para realizar el sistema programando un procesador general, pues bastará asociar los registros empleados en la descripción del sistema con los registros y posiciones de memoria del procesador, y traducir las operaciones a realizar al lenguaje de procesador.

Para diseñar una máquina especializada a partir de la descripción del algoritmo a nivel de transferencia de registros, hace falta descender, tal vez, a un nivel de descripción más detallado en el que sólo se empleen operaciones realizables por los circuitos lógicos que podemos emplear en el diseño del sistema (registros de desplazamiento, contadores, sumadores, puertas lógicas, etc.).

Hasta este punto el método de diseño propuesto coincide exactamente con el que se emplearía si se debiera realizar el

diseño programando un procesador general. Está por demostrar que la descripción es útil para diseñar el sistema con un circuito especializado.

10.2.3 Descripción de las operaciones de tratamiento de datos

En la descripción del sistema podemos distinguir dos tipos de proposiciones o sentencias, las de tratamiento de datos y las de control de secuencia.

Las primeras serán operaciones del tipo:

```
1:  A←A+B
2:  R←R+1
3:  C←C. AND.D
4:  CY_E (7..0)←E (7..0)_CY
```

Para realizarlas mediante un programa bastará transferir los datos a los acumuladores o registros susceptibles de ser operados y desdoblar las operaciones más complejas (por ejemplo suma o resta en coma flotante) en las operaciones realizables por la ALU del procesador.

El diseño de un circuito capaz de realizar estas operaciones no es tan inmediato, pues requiere una interpretación de las mismas.

La primera operación implica la existencia de circuitos sumadores, a cuyas entradas se conectarán los registros A y B, cuya salida se conecta al registro A. La segunda implica que R sea un contador susceptible de ser incrementado. La tercera, que haya puertas que realicen la operación AND entre los registros C y D. La cuarta, que E sea un registro de desplazamiento a la izquierda.

10.2.4 Descripción de las operaciones de control de secuencia

Estas operaciones se deducen de las sentencias de estructuración del algoritmo. Se puede demostrar que cualquier algoritmo se puede describir mediante sentencias secuenciales, como las de tratamiento, y sentencias de estructuración de cuatro tipos [1]:

```
5:  SI      A=1
        ENTONCES        <Sentencia si>
        SI NO           <Sentencias no>
    FINSI
6:  SI      N
        CASO 1          <Sentencia 1>
        CASO 2          <Sentencia 2>

        CASO m          <Sentencia m>
        SI NO           <Sentencia no>
    FINSI
7:      REPETIR
                        <Sentencias bucle>
        HASTA A=1
8:      MIENTRAS        A=1
        HACER           <Sentencias bucle>
        FINMIEN
```

Tanto en el caso de realizar el sistema con un programa como con un circuito, hay que desdoblar estas instrucciones de estructuración en sentencias del tipo:

```
8: SI A=1 ENTONCES IR A 18
```

Donde A, en general, será una expresión lógica, que relacionará los distintos bits de un registro; es el caso de los indicadores (flags) de las ALU de los microprocesadores; Z/NZ, P/M, PE/PO, o combinará los bits de varios registros entre sí (A=B, A>B).

Así, para traducir las sentencias de estructuración a lenguaje ensamblador haremos:

```
5: SI A=1                                  S5: SI A=1 ENTONCES IR A E5
     ENTONCES <Sentencias si>                    <Sentencias no>
     SI NO <Sentencias no>                       IR A F5
   FINSI                                   E5: <Sentencias si>
            ...                            F5: ......

6: SI N                                    S6: SI N=1 ENTONCES IR A C1
     CASO 1 <Sentencias 1>                     SI N=2 ENTONCES IR A C2
     CASO 2 <Sentencias 2>                           ....
            ...                                SI N=m ENTONCES IR A Cm
     CASO m <Sentencias m>                       <Sentencias no>
     SI NO <Sentencias no>                 C1: <Sentencias 1>
   FINSI                                       IR A F6
            ...                            C2: <Sentencias 2>
                                               IR A F6
                                                     ...
                                           Cm: <Sentencias m>
                                           F6: ....

7: REPETIR                                 R7: <Sentencias bucle>
     <Sentencias bucle>                        SI A≠1 ENTONCES IR A R7
   HASTA A=1                                         ...

8: MIENTRAS A=1                            M8: SI A≠1 ENTONCES IR A F8
   HACER <Sentencias bucle>                      <Sentencias bucle>
   FINMIEN                                       IR A M8
   ...                                     F8: ...
```

Para traducir las sentencias a un microprograma procederemos de la misma forma, es decir, haciendo que una variable o función *A*, afecte al cálculo de la dirección de la siguiente microinstrucción. Si el sistema se desea realizar sin un microprograma, es decir por el método tradicional de las máquinas secuenciales (modelos de Mealy y Moore [4]) basado en el cálculo de los estados internos del sistema [5], no hay más que asociar el registro de direcciones a los biestables de estado interno, y sustituir la ROM del microprograma por puertas lógicas, que realicen las funciones equivalentes a cada bit de la ROM.

Como vemos es posible aprovechar una descripción del sistema a nivel de transferencias entre registros y sentencias de estructuración, para realizarlo por cualquiera de los tres procedimientos considerados.

10.3 ESTUDIO DE LA INTERCONEXION CON UN CASSETTE ANALOGICO

10.3.1 Descripción del problema

El sistema a diseñar debe registrar en una cinta magnética el contenido de una zona de la memoria del microcomputador, comprendida entre dos direcciones dadas, dirección baja$=A_b$ y dirección alta $=A_a$.

El formato a emplear en la grabación consta de dos aspectos. La codificación de los datos a nivel de octeto, para consegir una cierta capacidad de detección de errores y de recomposición de los octetos originales a partir de la información registrada en serie en la cinta. La codificación de cada uno de los bits, mediante un procedimiento compatible con las características físicas de la cinta magnética, considerando especialmente el reducido ancho de banda que admite.

Para la codficación de los caracteres se emplea un código similar a los empleados para la perforación de cinta de papel.

Aquí vamos a describir la norma propuesta en un symposium organizado por la revista norteamericana BYTE en la ciudad de Kansas (USA) en Noviembre de 1975. Esta norma ha sido adoptada por la firma Motorala en sus equipos de enseñanza más económicos (KIT II).

Los datos se registran agrupados en bloques de tamaño relativamente grande (256 octetos). El conjunto de la grabación va precedido de treinta segundos de octetos no significativos (FF), para permitir a la cinta que estabilice su velocidad, y dejar pasar, eventualmente, el principio de la cinta en el que no se puede registrar. Al final del último bloque se registra el código ASCII de la letra «G» para indicar que es el final de la grabación.

Cada bloque está formado por una cabecera, los datos y una cola. La cabecera la constituyen el código ASCII de la letra «B», un octeto con el número de datos registrados en el bloque y dos octetos con la dirección en la que se debe almacenar el primer dato del bloque. La cola la constituyen 25 caracteres no significativos (FF), destinados a separar los bloques el tiempo necesario para que el lector pueda identificar el inicio del bloque siguiente, aún en el caso de que se hubiera perdido algún carácter al leer el bloque anterior.

Los bits de un octeto se registran comenzando por el de peso más bajo (LSB). Si el bit es un «uno» lógico «marca», se registran ocho períodos de un tono de 2.400 Hz. Si el bit es un «cero» «espacio», se registran cuatro períodos de un tono de 1.200 Hz. Para conseguir que los bits leídos sean agrupados en octetos, de la misma forma en que lo estaban al registrarlos, a los ocho bits del dato se añaden tres bits: un «espacio» de inicio antes de los bits de datos (start) y dos «marcas» de parada después. Este método de sincronización es el mismo que se emplea en las comunicaciones serie con los terminales alfanuméricos, lo que permite emplear una UART para la serialización. Entre carácter y carácter se registra la frecuencia de la marca, lo cual permite al circuito receptor sincronizarse al recibir el espacio de inicio de cada carácter.

El sistema de interconexión entre el microcomputador y el magnetofón tendrá dos partes completamente independientes: el registrador y el cargador. El primero se encargará de registrar en la cinta magnética los octetos almacenados en memoria. El cargador se encargará de almacenar en memoria los datos leídos en la cinta. Aquí vamos a detallar sólo la primera parte, entendiéndose que el procedimiento de diseño del cargador es similar.

De la descripción del formato de grabación se deduce que el algoritmo a ejecutar por el sistema será, en una primera aproximación:

```
     REGISTRAR UNA CINTA
1:   Leer las direcciones que limitan la zona de memoria a copiar: DIR.
     BAJA y DIR. ALTA.
2:   Registrar la cabecera de la grabación.
       REPETIR
3:         Registrar bloque
       HASTA dirección del último carácter registrado = DIR.ALTA
4:   Registrar el final de la grabación.
     FIN
```

10.3.2 Descomposición descendente del sistema

Ahora hay que detallar más cada uno de estos procesos en los que hemos descompuesto el problema general. El proceso 1 está directamente relacionado con la estructura del microcomputador en el que se vaya a ejecutar, y en cualquier caso será un sencillo problema de programación con la siguiente estructura:

```
1:    LEER LAS DIRECCIONES QUE LIMITAN LA ZONA A COPIAR
1.1:    DIR.BAJA←Dirección leída del terminal.
1.2:    DIR.ALTA←Dirección leída del terminal.
      FIN.
```

El segundo proceso, como todos los que implican registrar algo en la cinta, supone una preparación de los datos a registrar y la llamada a una rutina encargada de hacer el registro, que será la misma para todos los casos. Los procesos se desdoblan de la forma siguiente:

```
2:    REGISTRAR LA CABECERA DE LA GRABACIÓN
2.1:    CONTADOR←1024; Inicializa el contador de caracteres a registrar.
2.2:    Registrar banda de separación
      FIN.

3:    REGISTRAR BLOQUE
3.1:    Calcular el tamaño del bloque
3.2:    Registrar la cabecera del bloque
3.3:    Registrar los datos
3.4:    Registrar la cola del bloque.
      FIN.

4:    REGISTRAR FINAL GRABACION
4.1:    CARACTER←«G»
4.2:    Registrar CARACTER
      FIN.

5:    CALCULAR EL TAMAÑO DEL BLOQUE
5.1:    SI DIR.ALTA−DIR.BAJA≥256
5.2:      ENTONCES CONTADOR←256
5.2:    SINO CONTADOR←DIR.ALTA-DIR.BAJA+1
        FINSI
      FIN
6:    REGISTRAR CABECERA BLOQUE
6.1:    CARACTER←«B»             ; Registra el carácter «B»
6.2:    Registrar CARACTER
6.3:    CARACTER←CONTADOR        ; Registra n.º de octetos de datos
6.4:    Registrar CARACTER
6.5:    CARACTER←DIR.BAJA.A      ; Registra el octeto
                                 ; alto de la dirección.
6.6:    Registrar CARACTER
6.7:    CARACTER←DIR.BAJA.B      ; Registra el octeto
      FIN                        ; bajo de la dirección

7:    REGISTRAR LOS DATOS
        REPETIR
7.1:      CARACTER←(DIR.BAJA)    ; Registra el carácter contenido en
7.2:      Registrar CARACTER     ; la pos. mem. apuntada por DIR.B.
7.3:      DIR.BAJA←DIR.BAJA+1
7.4:      CONTADOR←CONTADOR−1
        HASTA CONTADOR=0
      FIN

8:    REGISTRAR LA COLA
8.1:    CONTADOR←25
8.2:    Registrar banda de separación
      FIN

9:    REGISTRAR BANDA DE SEPARACION
9.1:    CARACTER←OFFH            ; Registrará la constan-
        REPETIR                  ; te OFF en hexadecimal
                                 ; Tantas veces como
9.2:      Registrar CARACTER     ; indica el contador
9.3:      CONTADOR←CONTADOR−1
        HASTA CONTADOR=0
      FIN
```

En estas descripciones sólo aparecen operaciones sencillas, que es posible traducir a cualquier lenguaje ensamblador de forma automática. Las sentencias de estructuración se traducirán de acuerdo con las normas dadas en el apartado 10.2.4. Sólo queda por especificar la rutina de grabación de un carácter, que se puede descomponer de la forma:

```
10:   REGISTRAR CARACTER
10.1:   CONT.BIT←0
10.2:   BIT←0                          ; Registra bit de inicio
        REPETIR                        ; Desplaza sucesivamente
10.3:     Registrar BIT                ; en el biestable BIT los 8
10.4:     CARACTER_BIT←1_CARACTER      ; bits del CARACTER más
10.5:     CONT.BIT←CONT.BIT+1          ; los dos bits de parada con
10.6:   HASTA CONT.BIT=11              ; que se han ido llenando
      FIN                              ; los huecos en CARACTER
```

Estas operaciones se pueden realizar mediante una rutina en ensamblador o bien mediante un circuito especializado. De hecho la pastilla UART, en cualquiera de sus versiones, es capaz de realizar todas estas operaciones.

Para completar la descripción del sistema aún queda por desglosar el proceso de modulación del BIT que ha de ser registrado en la cinta. Ello implica registrar en la cinta un tono de una frecuencia determinada, para lo que hace falta disponer de un base de tiempo. Esta base de tiempo será un oscilador externo de 4.800 Hz, que dividida por 2 ó 4 nos dará señales cuadradas de 2.400 y 1.200 Hz.

Dado que mientras se calcula el siguiente carácter a enviar se debe seguir registrando el tono de la marca, conviene que al menos la operación de generar los tonos a registrar sea realizada por un circuito autónomo, para no tener que cronometrar cada segmento de programa. Por tanto vamos a emplear en la descripción de la tarea de registrar un bit, una sentencia consistente en «esperar el final de un ciclo de oscilador», lo cual significa que la operación siguiente se sincronizará con uno de los flancos de los impulsos producidos por el oscilador. Con todo lo dicho el desglose queda:

```
20: REGISTRAR BIT
20.1: TEMPORIZADOR←0                    ; Inicializa un contador de los
                                          períodos de tono a registrar.
      REPETIR
20.2:   SI BIT=0
          ENTONCES SALIDA←TEMPORIZADOR (1); bit peso 2
                                        ; Oscilador /4=1.200 Hz.
          SINO SALIDA←TEMPORIZADOR (0)  ; bit peso 1
                                        ; Oscilador/2=2.400 Hz.
        FINSI
20.3:   Esperar fin ciclo
20.4:   TEMPORIZADOR←TEMPORIZADOR+1
      HASTA TEMPORIZADOR=16             ; Completar 16 ciclos del
                                          oscilador=8 de 2.400 Hz.
                                          =4 de 1.200 Hz.
      FIN.
```

10.3.3 Descomposición en niveles

En la descripción de cualquier problema se aprecian una serie de niveles de realización, cada uno de los cuales engloba todos los procedimientos utilizados por el o los procedimientos de nivel superior [6].

Esta descomposición en niveles, previa a la realización, tiene la utilidad de mostrarnos los lugares por los que puede pasar la barrera de separación entre la parte del sistema a realizar por programas y la realizable por circuitos. Esta última incluirá los niveles más bajos, los más próximos al entorno exterior. La parte programada incluirá los procedimientos de los niveles superiores. Dentro de la parte programada, se puede incluso emplear un lenguaje de alto nivel (PL/M, MPL, PLZ, PASCAL) para los primeros niveles, y reservar el lenguaje ensamblador para los procedimientos de los niveles más bajos.

A la vista de los objetivos de esta descomposición, está claro que un procedimiento debe situarse en un nivel inferior o igual al más bajo de los ocupados por los procesos que lo usan, y en un nivel superior o igual al más alto de los ocupados por los procesos que usa. Ello es debido a que un programa puede desencadenar un proceso realizado mediante un circuito, de una forma natural, realizando una operación de E/S. En cambio, para que un circuito pueda desencadenar un proceso realizado por programa, hace falta recurrir a las interrupciones, lo cual en general no es deseable.

Las interrupciones complican la arquitectura del computador y se deben reservar para los casos de multiprogramación más complejos o para los procesos de E/S lentos, que consumen mucho tiempo en bucles de espera, y que conviene dejar aletargados para poder dedicar la UCP a otras tareas. En este caso, al finalizar la tarea realizada por circuito, se provoca una interrupción que despierta la tarea de nivel superior, que se encontraba aletargada.

Así en el problema que nos ocupa se aprecian los siguientes niveles, de superior a inferior (fig. 10.1).

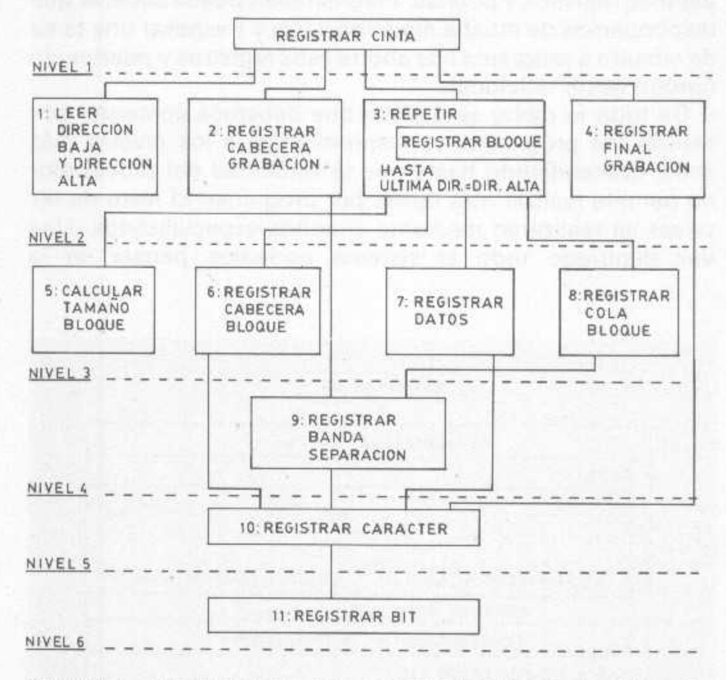


Figura 10.1 Tareas en las que se descompone el registro de una cinta audio, estructuradas en niveles de proximidad al circuito. Las tareas de los niveles más altos (1.2...) se realizarán por programa, y las de niveles más bajos (6.5...) mediante circuitos

1: Compuesto por el procedimiento principal o descripción más general del algoritmo.
2: Compuesto por los procedimeintos 1 a 4, pues son los empleados por el procedimiento del nivel 1.
3: Lo forman los procedimintos 5 a 8. Podríamos incluir en este nivel al procedimiento 9, pues es usado por los del nivel 2, pero como el 8 también lo usa, parece conveniente desplazarlo a un nivel inferior.
4: Formado por el procedimiento 9.
5: Contiene tan solo el procedimiento 10.
6: Contiene el proceso de nivel más bajo, el 20.

10.3.4 Realización

10.3.4.1 *Criterios de elección de la barrera programa/circuito*

En la realización debemos tener presente tres aspectos:
1) Tiempo de ejecución.
2) Economía de circuitos discretos.
3) Economía de memoria.

En primer lugar debemos satisfacer las restricciones de velocidad del entorno exterior, lo cual fijará un límite inferior de procedimientos realizables por circuito, pues es evidente que cualquier operación se realiza más rápidamente con un circuito especializado que con una máquina de uso general.

Una vez determinado este límite inferior de la barrera

circuito/programa, hay que resolver el compromiso entre memoria y circuitos externos.

Subir la barrera significa realizar más tareas con circuitos, lo cual implica el uso de circuitos más complejos y más caros, y que el sistema será menos flexible y modificable. Por contra tiene la ventaja de ahorrar memoria.

En los microcomputadores actuales, los incrementos mínimos de memoria suelen ser de 1 K o más, lo cual puede hacer que traspasar una tarea de programa a circuito nos ahorre un CI de memoria, a costa de añadir al circuito algunos registros y puertas. Pero también puede suceder que dispongamos de mucha memoria vacía y traspasar una tarea de circuito a programa nos ahorre esos registros y puertas sin ningún gasto adicional.

De todo lo dicho se deduce que debemos comenzar por realizar los programas correspondientes a los niveles más altos, descendiendo hasta que la velocidad del procesador no permita realizar más tareas por programa. El resto de las tareas se realizarán mediante circuitos especializados. Una vez depurado todo el sistema podemos pensar en la posibilidad de ahorrar algún banco de memoria traspasando tareas de programa a circuito, sin que ello suponga realizar un paso atrás, pues ya hemos visto que la traducción de la descripción a programa era automática, y además hemos tenido la posibilidad de simular el algoritmo diseñado, pudiendo detectar y corregir los errores de diseño sin necesidad de modificar los circuitos.

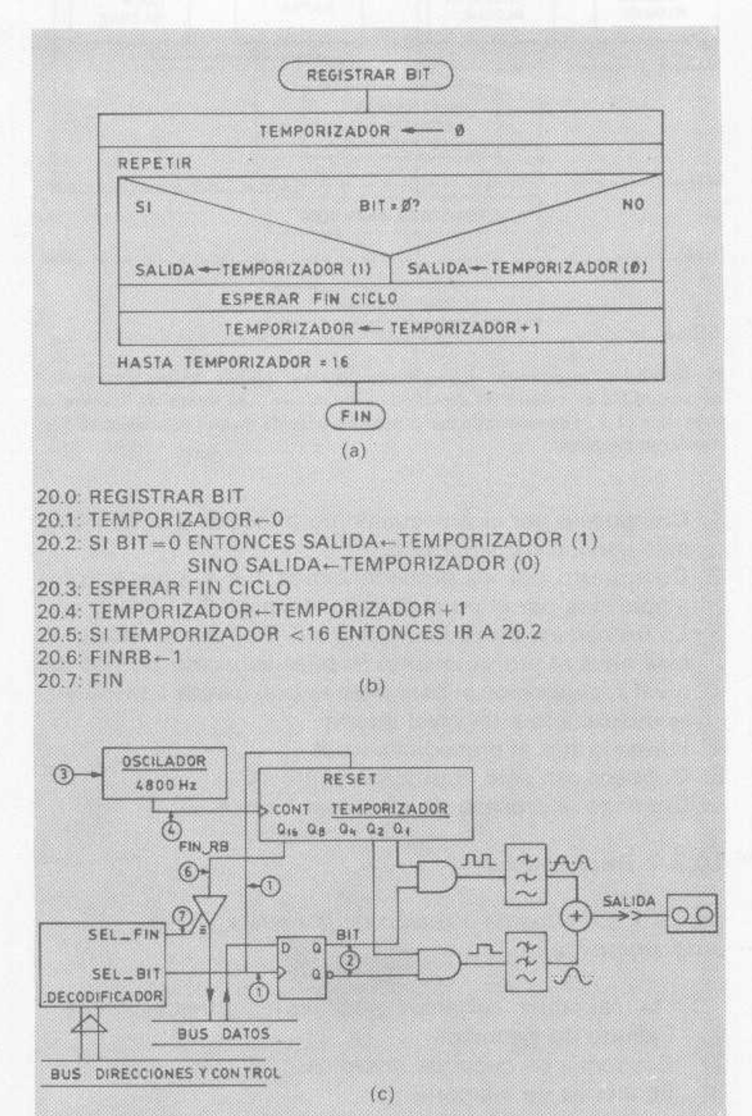


Figura 10.2 Descripción de la tarea de registrar un bit. a) En forma de organigrama de Nassi-Shneidermann. b) En lenguaje de transferencia de registros (RTL). c) En forma de circuito. Obsérvese que la operación 20.5. que debería detener el ciclo de registro de un bit, no se refleja en el circuito, ello se debe a que se ha tenido en cuenta la especificación de la norma que indica que en el período entre bits se debe seguir grabando

En el sistema que nos ocupa el tiempo de ejecución no es una limitación, pues el período de la señal a registrar es relativamente grande comparado con el del reloj del procesador. Ello significa que podríamos realizar todo el sistema mediante programas.

Según lo dicho, el procedimiento de diseño seguirá ahora con la realización de los programas correspondientes a cada tarea, comenzando por las de nivel más alto. Sin embargo, como el objetivo de este texto no es enseñar a programar, vamos a saltarnos esa fase del diseño y realizaremos los procesos por circuitos, comenzando por los de nivel más bajo.

10.3.4.2 *Tarea de registrar un bit*

Primeramente diseñaremos un circuito que realice el proceso de registrar un bit, basándonos en la descripción de la tarea 20. Tal como se ha dicho en el apartado 10.3.2, la sentencia «Esperar fin ciclo» supone la existencia de un oscilador de 4.800 Hz. Las sentencias 20.1, 20.4 y 20.5 describen el funcionamiento de un contador binario de 5 bits denominado TEMPORIZADOR. La sentencia 20.2 indica que la SALIDA del sistema está multiplexada siendo igual al bit 2^1 del TEMPORIZADOR si BIT=0, o al bit 2^0 si BIT=1

Ello conduce al circuito de la figura 10.2, en el que la simple escritura de un nuevo dato en BIT desencadena el proceso. Si nos fijamos en la descripción de la tarea que usa a ésta, veremos que no es una restricción, pues siempre se llama a esta tarea después de modificar BIT. Además la tarea de registrar un carácter termina registrando una marca (1), por lo cual durante el tiempo entre caracteres se seguirá registrando la frecuencia de la marca, tal como piden las especificaciones.

Para poder sincronizar las tareas de registrar un carácter y de registrar un bit, ésta debe producir una señal indicadora del final del proceso. Esta señal es la misma que indica el final del bucle, es decir el bit de peso 16 del TEMPORIZADOR.

10.3.4.3 *Tarea de registrar un carácter*

Veamos ahora la realización de la tarea de registrar un carácter mediante un circuito. De la descripción 10 se deduce que hacen falta un contador de 4 bits (CONT. BIT) y un registro de desplazamiento de 8 bits denominado CARACTER (8). El circuito que realiza esta tarea es el de la figura 10.3.

Una solución alternativa a este diseño consiste en emplear una UART o equivalente.

La realización de esta tarea mediante un circuito y no por programa, como se suponía cuando se realizó la tarea de registrar un bit, significa que la comunicación de esta tarea con la inmediata superior no se hace a través del bus del microcomputador, sino uniendo las señales homónimas del circuito que realiza la tarea superior.

Y así se podría seguir hasta realizar todas las tareas del sistema mediante circuitos.

Primeramente vamos a traducir la descripción de la tarea a ensamblador. De esta manera veremos la semejanza de las dos descripciones, RTL y ensamblador, así como la forma de

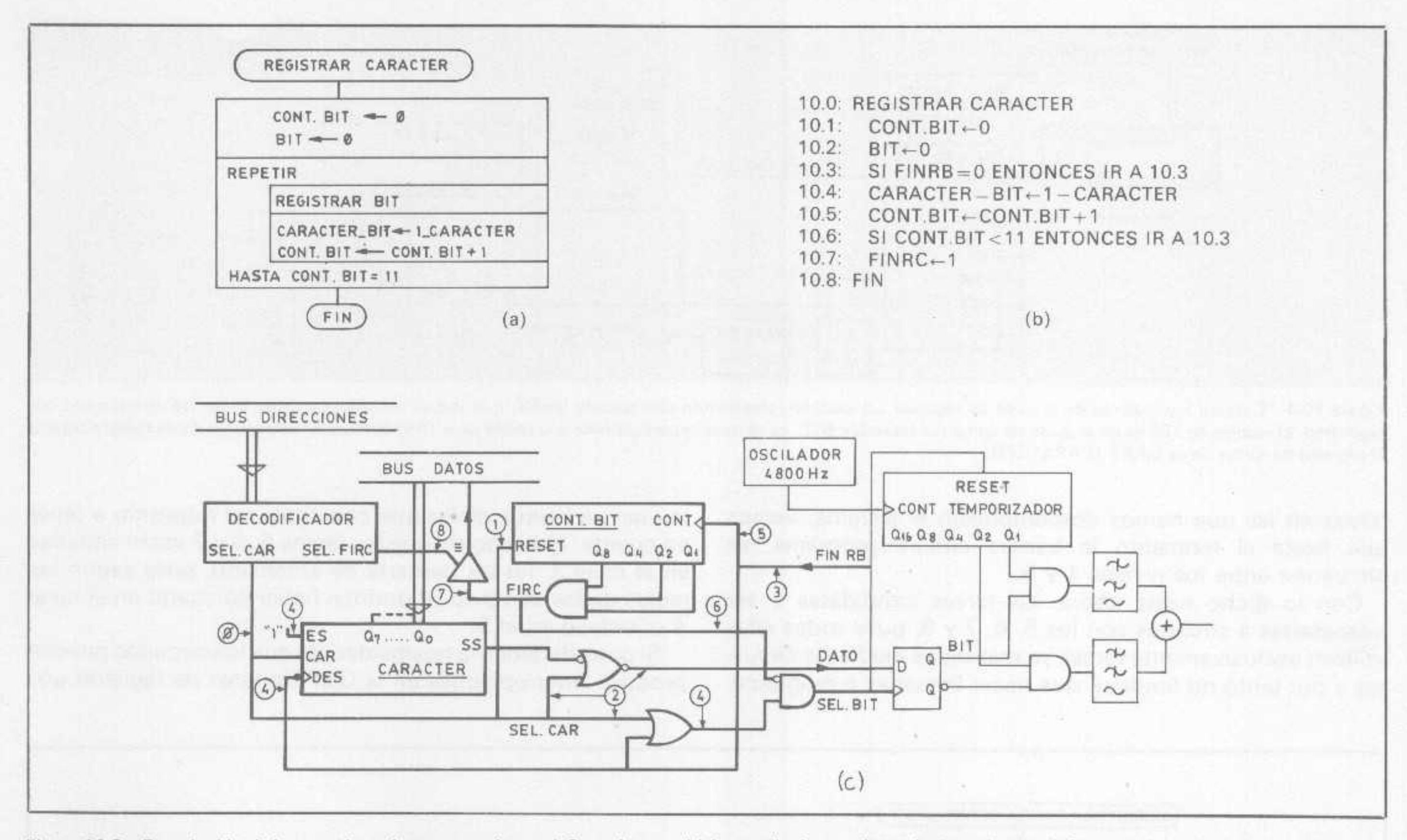


Figura 10.3 Descripción de la tarea de registrar un carácter. a) Organigrama. b) Lenguaje de transferencia de registros. c) Circuito. Además de la tarea descrita, se ha incluido el circuito de la figura 10.2 (en trazo más fino). Ello muestra las modificaciones necesarias para que interaccionen ambos circuitos, en vez de interaccionar con un programa a través del bus

sincronización entre programa y circuito. Después estudiaremos el circuito equivalente.

10.3.4.3.1 Solución programada

Del análisis de los datos empleados por la rutina vemos que sólo usa un biestable BIT, y los registros denominados CONT. BIT y CARACTER. El primero es un contador de 0 a 11, con posibilidad de ser puesto a cero. CARACTER es un registro de 8 bits, que almacena el carácter a registrar. Suponiendo que el microprocesador cuenta con al menos, tres registros, asociamos CONT.BIT al registro R1 y CARACTER al R2. Suponiendo que el biestable BIT de la tarea de registrar un bit está conectado al bit 7 del bus de datos, y que FINRB también se lee a través del bit 7, la descripción de la tarea en un ensamblador cualquiera podría ser:

```
      LOAD       R1, # 0      ; CONT.BIT←0
      LOAD       R0, # 0      ; BIT←0
                              ; REPETIR
B1:   OUT        BIT, R0      ; REGISTRAR BIT
B2:   INP        R0, FINRB    ; Espera hasta que
      AND        R0, # 80 H   ; Bit 7 de R0=1 (FINRB=1)
      JUMP, Z    B2
                              ; CARACTER _ BIT ← 1 _ CARACTER
      SET        CY           ; Pone a 1 el bit de carry de la UAL
      RRC        R2           ; Rota hacia la derecha CARACTER
                              ; y CARRY.
      JUMP, CS   F1           ; Guarda en bit 7 de R0 el CARRY
      LOAD       R0, # 0      ; Si CARRY=1 no hay que
                              ; hacer nada porque R0 (7)=1
F1:   INC        R1           ; CONT.BIT←CONT. BIT+1
      COMP       R1, # 11     ; HASTA CONT.BIT=11
      JUMP, NZ   B1
      RETURN                  ; FIN.
```

En el campo de los comentarios se han incluido las sentencias de la descripción en RTL, para hacer más patente la correlación entre ambas.

10.3.4.3.2 Solución cableada

El circuito equivalente a este programa tendrá los mismos registros, salvo el R0, que se empleaba para comunicar el programa con el circuito, y que ahora no es necesario. La figura 10.3*c* nos muestra una posible realización.

Las líneas de control se han marcado con un número inscrito en un círculo, que corresponde al número de la sentencia RTL que realiza esa línea de control.

Al cargar el registro CARACTER (10.0), SEL.CAR=1, se desencadena la tarea, poniéndose a 0 el contador CONT.BIT (10.1) y almacenándose un 0 en el biestable BIT de la tarea de registrarlo (10.2).

Obsérvese que para ello se ha «multiplexado» la señal SEL.CAR negada, con la salida serie (SS) del registro de desplazamiento CARACTER. Al finalizar el registro del BIT, FINRB=1, se incrementa CONT.BIT (10.5), se desplaza CARACTER, colocando un 1 en el bit más significativo (10.4) y almacenando en BIT el bit menos significativo de CARACTER, reiniciándose así el ciclo. Al llegar CONT.BIT a 11, FIRC se hace 1, lo que permite al programa detectar al fin de la tarea de registrar el carácter.

La figura 10.4 muestra otra versión de la tarea en la que se ha empleado una UART para serializar los datos.

10.3.4.4 *Registrar la banda de separación*

Si observamos la figura 10.1, en la que se encuentran las

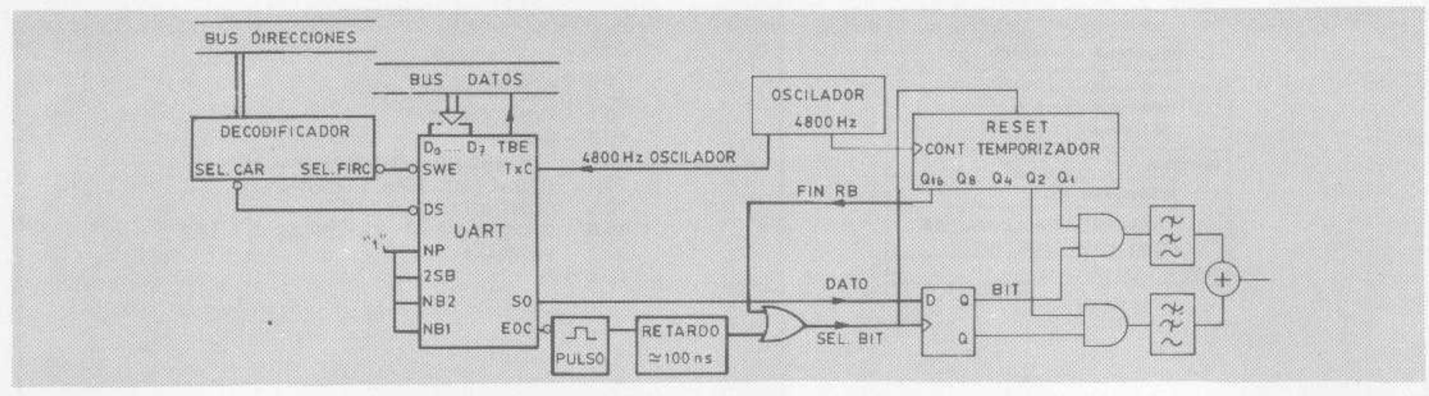


Figura 10.4 Circuito equivalente de la tarea de registrar un carácter, empleando una pastilla UART, que realiza automáticamente todas las operaciones del algoritmo. El retardo de 100 ns en el pulso de carga del biestable BIT., es necesario para permitir a la salida serie (S0) cambie de valor después de haber cargado el registro de datos de la UART (CARACTER)

tareas en las que hemos descompuesto el sistema, vemos que hasta el momento la barrera circuito/programa, se encuentra entre los niveles 4 y 5.

Con lo dicho hasta ahora, las tareas candidatas a ser traspasadas a circuitos son los 5, 6, 7 y 9, pues todas ellas utilizan exclusivamente tareas ya realizadas mediante circuitos y por tanto no tendrían que hacer llamadas a programa, lo cual ya hemos dicho que constituía un requisitio a tener en cuenta. El hecho de que las tareas 5, 6 y 7 están situadas en el nivel 3, no las descarta de antemano, pues según las reglas dadas, también se podrían haber colocado en el nivel 4 o incluso en el 5.

Si consideramos la posibilidad de que los circuitos puedan producir interrupciones en la UCP, la tarea de registrar una

REGISTRAR BANDA SEPARACION
CARACTER ← ØFFH
REPETIR
REGISTRAR CARACTER
CONTADOR ← CONTADOR − 1
HASTA CONTADOR = Ø
FIN

(a)

9.0: REGISTRAR BANDA SEPARACION
9.1: CARACTER←0FFH
9.2: SI FIRC=0 ENTONCES IR A 9.2
9.3: CONTADOR←CONTADOR-1
9.4: SI CONTADOR≠0 ENTONCES IR A 9.2
9.5: FIRS←1
9.6: FIN

(b)

(c)

Figura 10.5 Descripcion de la tarea de registrar la banda de separacion. a) Diagrama de Nassi-Sheidermann b) Descripción en RTL. c) Circuito equivalente. La UART puede recibir datos del bus o bien la constante FF (hexadecimal), mediante un multiplexor (MXDR)

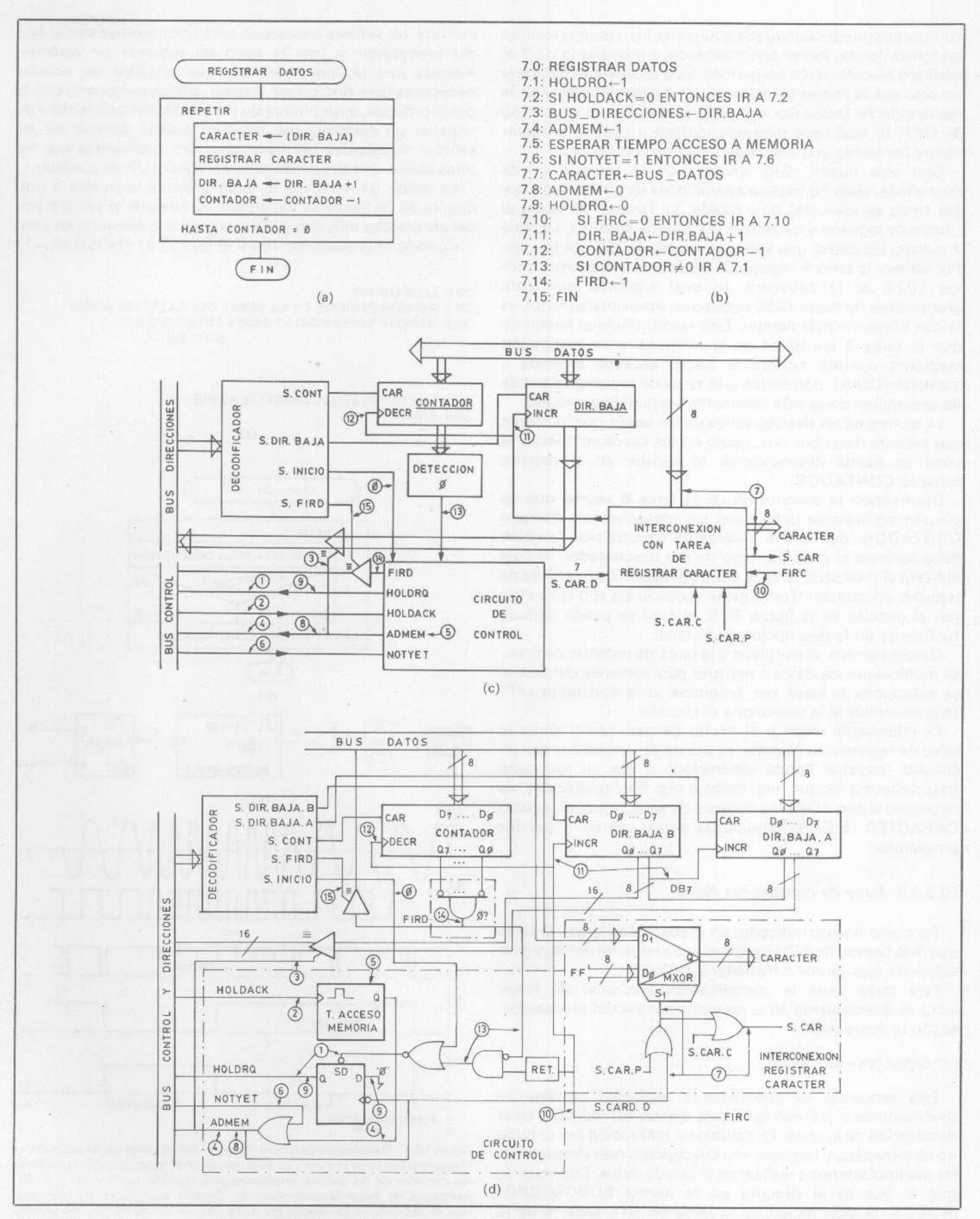


Figura 10.6 Descripcion de la tarea de registrar datos. a) Organigrama de Nassi-Shneidermann. b) Descripción RTL. Obsérvese que se han explicado con mayor detalle las operaciones de petición y liberación del bus para acceder a la memoria (7.1 a 7.9). c) Diagrama de bloques. d) Circuito lógico equivalente. Se han encuadrado las partes que corresponden a cada bloque. La señal de inicio generada por el decodificador, podría hacerse coincidir con cualquiera de las tres que seleccionan la carga de los registros. Por ejemplo S. CONT. que siempre se deberá actualizar en ultimo lugar. El ciclo de acceso a memoria dura lo que indique el monostable «T. acceso memoria», salvo que durante ese tiempo se active la señal NOTYET (espera), indicando que la memoria es más lenta

cinta puede quedar aletargada, y mientras los circuitos realizan las tareas que les hayan encomendado, y mientras la UCP se dedica a ejecutar otros programas. Esta posibilidad hace que no sólo sea el ahorro de memoria el objetivo a cubrir con la realización de tareas por circuitos, sino el ahorro de tiempo de UCP, lo cual hace deseable traducir a circuito primeramente las tareas que consumen más tiempo de UCP.

Con este nuevo dato, vemos que la tarea 5 queda descartada, pues no implica operaciones de entrada/salida y por tanto se ejecutará muy rápida. La tarea 6 consume el tiempo de registrar 4 caracteres por bloque de datos. La tarea 7 registra los datos, que son un máximo de 256 por bloque. Por último, la tarea 9 registra 25 caracteres por bloque, más los 1024 de la cabecera, lo cual significa que para grabaciones de hasta 1250 caracteres, aproximadamente, es la que consume más tiempo. Esta razón, unida al hecho de que la tarea 7 lee datos en la memoria y su realización mediante circuito supondría hacer accesos directos a memoria (DMA), convierten a la tarea de registrar la banda de separación en la más fácilmente traducible a circuito.

La lectura de las descripciones de las tareas que la llaman nos permite descubrir que, como en los casos anteriores, la tarea se puede desencadenar al escribir en el registro llamado CONTADOR.

Observando la descripción de la tarea 9 vemos que su circuito equivalente debe tener un «descontador» llamado CONTADOR, de 10 bits y cargable en paralelo. También debe detectar al paso por cero de ese descontador, lo cual indicará el final de la tarea, y debe además activar la tarea de registrar un carácter. Todas estas operaciones son realizadas por el circuito de la figura 10.5, el cual se puede deducir fácilmente de la descripción de la tarea.

Obsérvese que, al acoplarla a la tarea de registrar carácter, se multiplexan los datos a registrar para tomarlos del bus, si se selecciona la tarea por programa, o la constante «FF» (hexadecimal) si la selecciona el circuito.

Es interesante recalcar el hecho de que, como ahora la tarea de registrar un carácter, se puede desencadenar por un circuito (registrar banda separación) o por un programa (reg. cabecera bloque, reg. datos o reg. final grabación), es necesario sumar (OR) las órdenes de seleccionar el registro CARACTER (S.CAR) producidas por el circuito y por los programas.

10.3.4.5 *Tarea de registrar los datos*

Tal como hemos indicado, en grabaciones largas la tarea que más tiempo de CPU ocupa es la de registrar datos, y es la siguiente que vamos a trasladar a circuito.

Esta tarea tiene la complicación adicional de hacer lecturas directamente en la memoria central del procesador, según la instrucción:

7.1: CARACTER←(DIR.BAJA).

Esta sentencia es directamente realizable en algunos procesadores y por ello la hemos dejado así indicada en la descripción de la tarea. En cambio su realización por circuito no es inmedata, y requiere una descripción más detallada de las manipulaciones a realizar en el bus de datos. Suponiendo que el bus es el descrito en la norma EUROMICRO-MUBUS, la descripción de la tarea en RTL sería la de la figura 10.6*b*.

El circuito que corresponde a esta descripción es el de la figura 10.6*c*. En ella vemos que el circuito consta de dos registros contadores y un bloque de control, encargado de producir las señales necesarias para sincronizarse con el bus del procesador y con la tarea de registrar un carácter. Además hay un decodificador, que produce las señales necesarias para sincronizar la tarea con los programas que la desencadenan, y un bloque de interconexión con la tarea de registrar un carácter, que tiene en cuenta además de las señales producidas por esta tarea, las producidas por las otras tareas que desencadenan el registro de un carácter.

La orden de inicio de la tarea provoca la puesta a uno directa de un biestable encargado de solicitar el uso del bus del procesador mientras dura el acceso a la memoria de ésta.

Cuando el procesador libere el bus (7.2) (HOLDAK=1)

30.0: LEER UN BIT
30.1: REPETIR PERIODO DE LA SEÑAL DEL CASSETTE AUDIO
30.2: REPETIR SI PERIODO>1/1800 s ENTONCES BIT←1
SINO BIT←0

30.3: HASTA FIN SEÑAL CASSETTE AUDIO
30.4: FIN

(a)

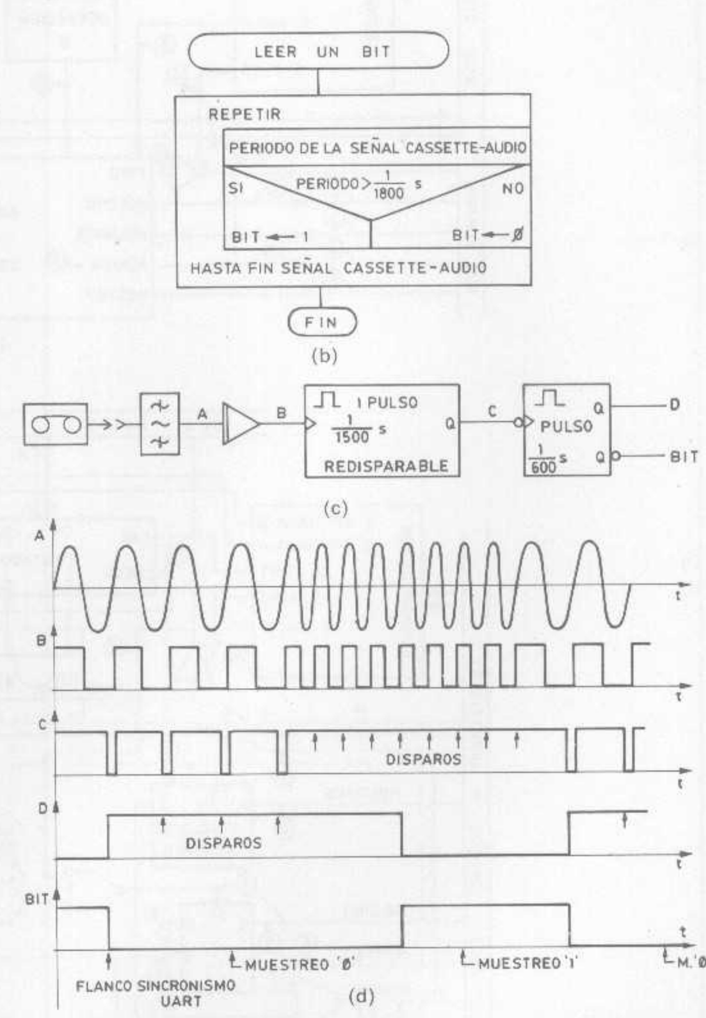


Figura 10.7 Descripción de la tarea de leer un bit. a) Lenguaje de alto nivel. La discriminación entre el 0 y el 1 se hace comparando el período con la media de los períodos de las señales empleadas para registrar un 0 y un 1. b) Organigrama de Nassi-Shneidermann. c) Circuito equivalente. d) Cronogramas de las señales generadas por cada uno de los elementos del circuito. Obsérvese que la duración del «1» lógico es menor que la del «0», ésto solo acontece cuando el 1 sigue a un 0, ya que el último monoestable debe generar un pulso de duración doble a la del período de la señal «0», para cubrir la posible pérdida de un impulso en la señal C, por problemas de ruido o fluctuaciones de la red

se inicia el acceso a memoria, transfiriendo el contenido del registro de dirección al bus (7.3) y activando la señal ADMEM (7.4). Este acceso durará al menos un ciclo del procesador, el cual viene fijado por un monoestable, (7.5) y si la memoria es lenta y activa la señal NOTYET para alargar el ciclo de acceso, éste durará hasta que esa señal se desactive (7.6). Una vez finalizado este ciclo de espera, se carga el registro CARACTER (7.7) y se finaliza el acceso a memoria (7.8-7.9). Al cargar CARACTER se inicia el ciclo de escritura de un carácter, y el circuito espera el fin de este ciclo, FIRC (7.10), el cual desencadena la actualización del registro de direcciones (7.11) y del contador (7.12).

La repetición de estas operaciones, mientras el registro CONTADOR es distinto de cero, se consigue haciendo que la señal indicativa del final del ciclo FIRC $=1$ y CONTADOR$\neq 0$, FIRD$=0$ (7.13), provoquen una nueva señal de INICIO (7.1). Para evitar el impulso espurio que se produciría inmediatamente antes de ponerse a 0 el contador, el cual provocaría un ciclo más de los deseados, se debe retardar (RET) la señal FIRC antes de combinarla con FIRD.

El control del estado de la tarea por el programa que la desencadenó se hace mediante la lectura de la señal FIRD, que actúa como bit de estado.

10.3.4.6 *Tarea de leer un bit*

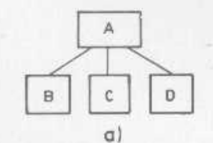
Una vez apuradas todas las posibles versiones del circuito de grabación, vamos a analizar la tarea más baja del circuito de interpretación de la cinta. La descripción de esta tarea se da en la figura 10.7, la cual muestra una posible realización de la tarea empleando monostables.

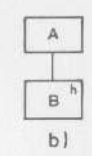
La señal analógica del cassette se «digitaliza» con un amplificador «trigger-Schmitt».

El primer monostable produce impulsos de 1/1800 s, si el período de la señal de excitación es mayor, o un 1 constante si el período es menor, pues los flancos ascendentes producen un nuevo impulso antes de que se acabe el anterior, solapándose ambos. Dado que la presencia de impulsos negativos a la salida del primer monostable indica que se está recibiendo un 0, se emplean los flancos de bajada para producir un impulso suficientemente largo como para que se solapen, produciendo un solo impulso durante toda la duración de un 0, o hasta que comience un 1. La determinación de si el bit leído es 0 o 1, también se puede hacer mediante un circuito lineal llamado PLL, (Phase Locked Loop = bucle enganchado en fase). Estos circuitos son discriminadores de frecuencia, que proporcionan una tensión proporcional a la frecuencia de la señal de entrada, dentro de unos márgenes, y con una sensibilidad controlable mediante componentes discretos.

10.4 FORMATOS DE GRABACION

Vamos a describir los procedimientos de grabación sobre cassettes audio más conocidos. Para ello vamos a emplear,





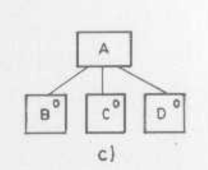


Figura 10.8 Método de Jackson para la descripción de ficheros. a) El bloque A se compone de una sucesión de los bloques B, C y D en este orden. b) El bloque A se compone de la repetición *n* veces del bloque B. Si *n* se desconoce se indica con *, c) El bloque A se compone de uno de los tres bloques B, C o D, según los casos

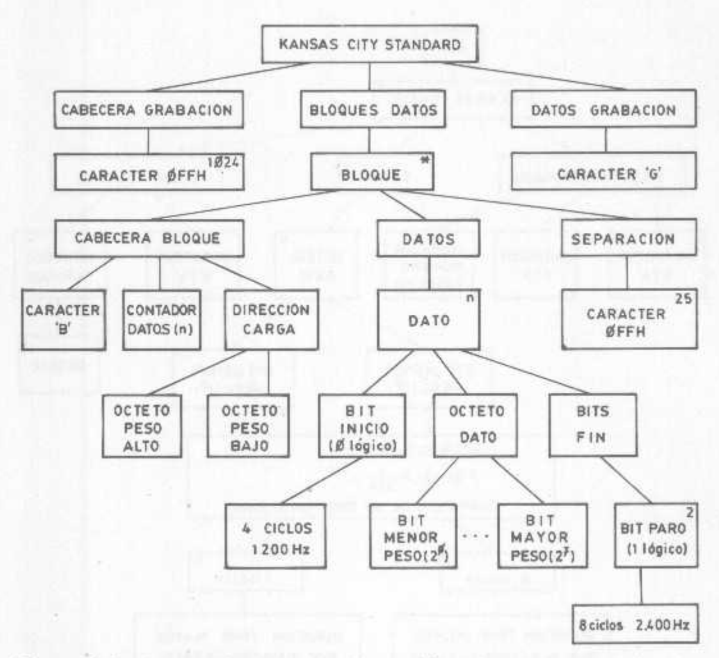


Figura 10.9 Formato de grabación «Kansas City». Los caracteres indicados entre comillas (G,B) se registran codificados en ASCII

además de la lista de requisitos, una gráfica en la que se muestra la estructura final de un programa registrado. En esta gráfica se ha usado el método de Jackson para la descripción de ficheros, que se describe brevemente en la figura 10.8.

10.4.1 Norma «Kansas City Standard»

Es la descrita en este capítulo, y usada por Motorola en sus Kits de entrenamiento (KIT-II).

Aprobado por un symposium patrocinado por la revista BYTE en Kansas en noviembre de 1975. Sus características más importantes son:

1. 300 baudios.
2. 1 = «Marca» = 8 ciclos de 2400 Hz.
   0 = «Espacio» = 4 ciclos de 1200 Hz.
3. Carácter: Espacio = bit de inicio. (Start).
   —ocho bits de datos
   —dos «Marcas» = bits de parada (stop).
4. Entre caracteres se envía la frecuencia de la marca.
5. Se envía primero el bit de peso más débil (LSB) y al final el de peso fuerte (MSB).
6. Los datos se organizan en bloques de longitud arbitraria, *precedidos por cinco segundos de marcas,* como mínimo.
7. No se debe registrar información significativa en los primeros 30 segundos de cinta.

En el Kit II de Motorola se registra con el formato de bloques de la figura 10.9.

10.4.2 Formato HIT (Hobbyst Interchange Tape System)

Propuesto por Jerry Ogdin en la revista POPULAR ELECTRONICS de Sept. 75. Sus características más importantes son:

1) Hay un intervalo de bit fijo, que condiciona la velocidad de grabación entre 30 y 300 baudios.
2) El intervalo consta de un período de silencio precedido de un tono de 2 kHz. Si la duración del tono es mayor que la del silencio, el bit es un uno. Si la duración del silencio es mayor, el bit es un cero.

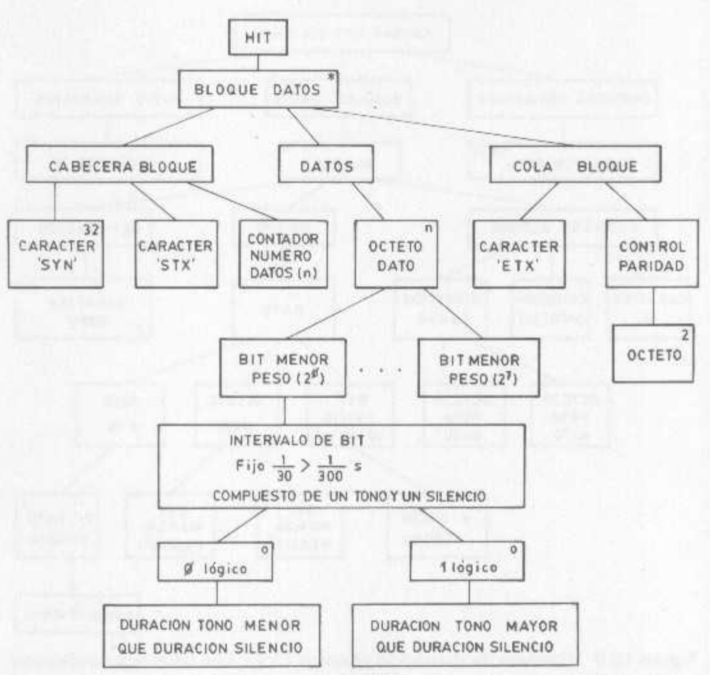


Figura 10.10 Formato de grabación «HIT». Para detectar el inicio de los bloques y el final correcto del bloque de datos, se emplean los caracteres ASCII: SYN = Sincronismo; STX = Inicio de texto y ETX = Fin de texto.

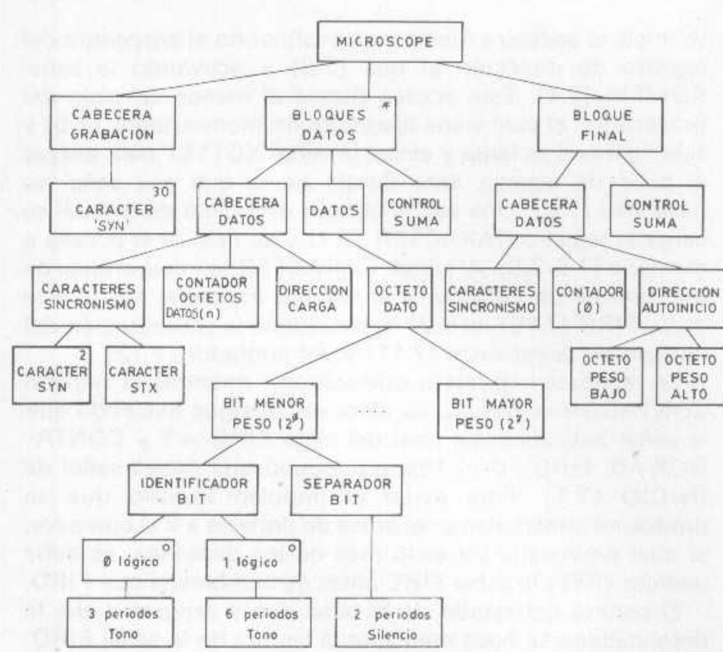


Figura 10.11 Formato de grabación MICROSCOPE-ELECLUB. Este formato tiene gran difusión, fundamentalmente en Suiza. Se caracteriza por emular los formatos octales de las cintas de papel, introduciendo la dirección de autoinicio. La decodificación de la señal reproducida es algo más compleja que la descrita en el capítulo, pero también es menos sensible a las fluctuaciones de velocidad de la cinta.

3) Se envía primero el bit de peso más débil.
4) Los bloques de datos se configuran de acuerdo con la figura 10.10.

10.4.3 Formato Eleclub-Microscope

Es un formato desarrollado en el laboratorio de calculadoras digitales de la Escuela Politécnica Federal de Laussanne (Suiza) en 1975. Este formato se basa en el «HIT», descrito anteriormente, con las siguientes diferencias (fig. 10.11):

1) Hay una cabecera de grabación de 30 caracteres SYN.
2) La cabecera de bloque sólo tiene 2 caracteres SYN.
3) La cabecera de bloque tiene además, dos octetos con la dirección a partir de la cual se deben cargar los datos de ese bloque.
4) El control de paridad es un solo octeto y es el complemento a dos de la suma de todos los octetos del bloque, incluidos el contador y la dirección (control de suma = check sum = CHS).
5) El carácter ETX se ha suprimido.
6) Hay un último bloque sin datos (contador = 0), cuya dirección es la de autoinicio del programa cargado, si es distinta de 1.
7) El 0 lógico se registra mediante tres ciclos de portadora (3 kHz) y dos períodos de silencio.
8) El 1 lógico se registra mediante seis ciclos de portadora y también dos períodos de silencio.

BIBLIOGRAFIA

[1] N. Wirth: «Systematic Programming: An Introduction». Prentice-Hall. 1973.
[2] J. Ogdin: «Hobbyist Interchange Tape System». Popular Electronics. SEP. 1975.
[3] Evaluation Kit II Manual. Motorola.
[4] Nagle y otros: «An Introduction to Computer Logic». Prentice-Hall. 1975.
[5] M.M. Mano: «Computer System Architecture». Prentice-Hall. 1976.
[6] J. Stockenberg & A. van Dam: «Vertical migration for Enhancement in layered Hardware/Firmware/Software Systems». COMPUTER IEEE de mayo de 1978.
[7] M. Medina & J.D. Nicoud: «Simple cassette interface» MICROSCOPE de febrero de 1977.
[8] Schneider, Rudy L.: «Build this SDK-85 cassette interface». EDN. 20-Mar-79.
[9] Wirth, Niklaus: «Algorithms + Data Structures = Programs». Prentice-Hall 1976.
[10] Hill, Chu, Dietmeyer, Sietwiorek: «Introducing Computer Hardware Description Languages (CHDL)». Computer IEEE. Dic-74.
[11] CHU, Y: «Computer Organization and Microprogramming». Prentice-Hall, 1972. (en castellano).
[12] Nassi-Shneidermann: «Flowchart Techniques for Structured Programming». ACM Sigplan vol. 8.8. Ago. 73.
[13] Jackson, M.A.: «Principles of program design». Academic Press. 1975.
[14] Dijkstra. E.W.: «A discipline of programming». Prentice-Hall Series in Automatic Computation. 1976.
[15] Bass, C.: «PLZ: A family of System programming microprocessors.» Computer IEEE, Mar. 78.
[16] Nichols, J.L.: «Hardware versus software for microprocessors I/O». Computer Design. Ago-76.

discos flexibles

11

Mateo Valero, José M.ª Pallarés

11.0 INTRODUCCION

Los discos flexibles constituyen uno de los periféricos para almacenar información más interesantes de los sistemas microcomputadores. Características que han influido notablemente en su aceptación y utilización han sido entre otras las de bajo precio, mayor fiabilidad que las cintas magnéticas y un tiempo de acceso a la información relativamente pequeño comparado con las velocidades de tratamiento de esa información por los microprocesadores. Los primeros discos flexibles fueron introducidos en el mercado por IBM en el año 1970, es decir, antes de que aparecieran los microprocesadores. El modelo se denominó 23FD y contenía un disco flexible de 8 pulgadas con una sola cara y con código de grabación en simple densidad. Posteriormente, en 1975, Shugart usó un código de doble densidad M^2FM en su modelo SA800 con lo que duplicó la cantidad de información almacenada. En el año 1976, la misma Shugart introdujo un disco flexible de 5 1/4 pulgadas que recibió el nombre de minidisco.

El capítulo está dividido en siete apartados. En el primero tratamos de las características de los discos, capacidades de almacenamiento y transportes físicos de los discos. En el siguiente describimos los códigos empleados para la grabación. En el tercero exponemos diferentes formatos de pista. En el cuarto tratamos brevemente de las funciones que deben ser implementadas entre el sistema microcomputador y el transporte del disco. En el siguiente describimos las características generales de algunos controladores integrados. Los dos últimos apartados describen la realización de dos interfaces concretas: la primera para simple densidad y la segunda para doble densidad.

11.1 CARACTERÍSTICAS DE LOS DISCOS

Los discos están constituidos por dos capas, una de ellas hace de sustrato y la otra, construida a base de un óxido magnético de aproximadamente 7 μm de espesor, que es donde se graba la información. Ambas capas están unidas y cerradas en un sobre de plástico. Como hemos dicho anteriormente, existen dos tamaños de discos estandarizados de 8 y 5 1/4 pulgadas.

En la figura 11.1 se muestra la forma de los discos así como los valores de algunas dimensiones.

Los discos flexibles se distinguen entre sí por el tamaño (8 y 5 1/4 pulgadas), el número de caras utilizables (1 ó 2), el código de grabación (simple o de doble densidad) y el formato de grabación (sectorizados por hardware o por software).

En los discos sectorizados por software existe un único agujero en el disco que al pasar por el agujero correspondiente, realizado en el sobre de plástico, es detectado por la circuitería de forma que define el principio de cada pista. Los

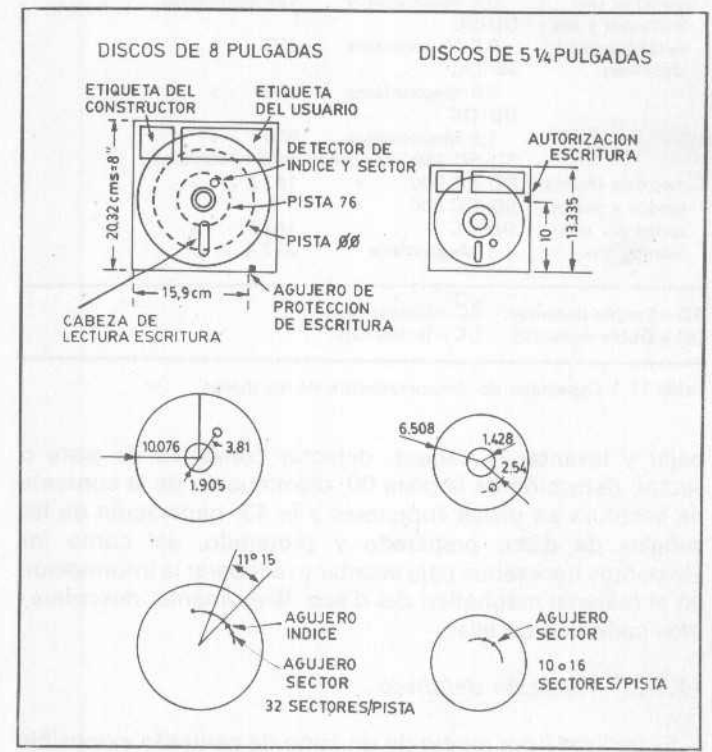


Figura 11.1 Dimensionado de los discos

discos sectorizados por hardware poseen un agujero físico por cada sector y dos agujeros próximos indicativos del principio de pista. El paso de dichos agujeros es detectado, normalmente, por el conjunto fotodiodo-fototransistor.

Existe sobre el disco una marca que se hace servir para protegerlo contra escritura; los discos de 8 pulgadas quedan protegidos al dejar la marca visible y con los minidiscos ocurre lo contrario.

11.1.1 Capacidad de almacenamiento

La capacidad de almacenamiento de los discos depende del tamaño, código de grabación, formato de grabación y si está sectorizado por hardware o por software. En la tabla 11.1 se resumen algunos valores típicos.

11.1.2 Transporte del disco

En la figura 11.2 se representan los transportes para discos de 8 pulgadas (SA800-801) y minidiscos (SA400) del fabricante Shugart.

En general, los transportes de los discos contienen la circuitería electrónica y elementos mecánicos necesarios para realizar las siguientes funciones: rotación del disco, movimiento de la cabeza de lectura-escritura entre pistas,

Característica	*Disco de 8 pulg.*	*De 5 ¼ pulg.*	*Comentarios*
Número de pistas	77	35 ó 40	
Densidad radial de las pistas	48 por pulgada	48 por pulgada	
Número de sectores (hardware)	32	10, 16 y 18	
Número de sectores (software)	8, 15, 26	8, 16, 18	
Densidad de grabación	SD: 3200 bits/pulg. DD: 6400 bits/pulg.	2580 b/p 5160 b/p	En la pista más pequeña
Número de caras	1 ó 2	1 ó 2	
Capacidad (sin formatear y sectorizados por software)	SD-SC 0,4 Megaoctetos DD-SC 0,8 Megaoctetos SD-DC 0,8 Megaoctetos DD-DC 1,6 Megaoctetos	 125 kilooctetos 250 » 250 » 500 »	
Capacidad (formateados y sectorizados por software)	SD-SC 250 kilooctetos DD-SC 500 » SD-DC 500 » DD-DC 1 Megaocteto	80,6 kilooctetos 161,2 » 161,2 » 322,4 »	

SD = Simple densidad; SC = Simple cara.
DD = Doble densidad; DC = Doble cara.

Tabla 11.1 Capacidad de almacenamiento de los discos

bajar y levantar la cabeza, detectar comienzo de pista o sector, detección de la pista 00, disminución de la corriente de escritura en pistas superiores a la 43, generación de las señales de disco preparado y protegido, así como los elementos necesarios para escribir y recuperar la información en el material magnético del disco. Brevemente, describiremos cada una de ellas.

11.1.2.1 *Rotación del disco*

Es realizada por medio de un cono de centrado extensible que se introduce en el agujero central del disco al cerrar la puerta del transporte. En los discos de 8 pulgadas la velocidad de rotación es de 360 vueltas/minuto y se emplean en la mayor parte de los transportes motores síncronos o asíncronos alimentados por la tensión de la red, si bien existe algún transporte que emplea motores de corriente continua. En los minidiscos sólo se emplean motores de corriente continua para obtener una velocidad de rotación del disco de 300 vueltas/minuto.

11.1.2.2 *Movimiento entre pistas*

Esta parte del transporte es la encargada de desplazar la cabeza de lectura-escritura sobre cada una de las pistas. Para ello casi todos los transportes tienen un motor paso a paso que realiza esta función a partir de las fases enviadas por el circuito controlador (no perteneciente al transporte). El tiempo empleado para pasar la cabeza de una pista a otra contigua suele estar entre 3 a 10 ms. Debido al movimiento lateral de la cabeza, ésta tarda un cierto tiempo en quedar centrada sobre la pista. Este tiempo varía normalmente entre 0 y 20 ms (siendo el valor 0 para los motores lineales).

11.1.2.3 *Activación de la cabeza sobre el disco*

Es el mecanismo que permite poner en contacto físico la cabeza de lectura-escritura con el material magnético del disco, de forma que éste también quede en un plano perpendicular con relación a la cabeza. Estas funciones son realizadas por un electroimán. La señal que dispara el electroimán proviene del circuito controlador del transporte.

11.1.2.4 *Detección de principio de pista y sector*

Los transportes poseen mecanismos que les permite detectar e informar al controlador de los instantes en los que los agujeros físicos correspondientes al comienzo de pista o comienzo de sector (en los discos formateados por hardware) pasan por el agujero correspondiente del disco. La detección del paso de un agujero se realiza a través de una pareja de fotodiodo y fototransistor. A partir de ahí existe un circuito sencillo que determina si lo que se detectó fue el principio de pista o de sector. Ambas señales de principio de pista y sector son enviadas hacia el controlador.

11.1.2.5 *Detección de la pista 00*

El hecho de que la cabeza está situada sobre la pista 00 (la más externa del disco) es detectado por una pareja fotodiodo-fototransistor o por algún interruptor. Cuando esto ocurre, el transporte envía la señal correspondiente al controlador.

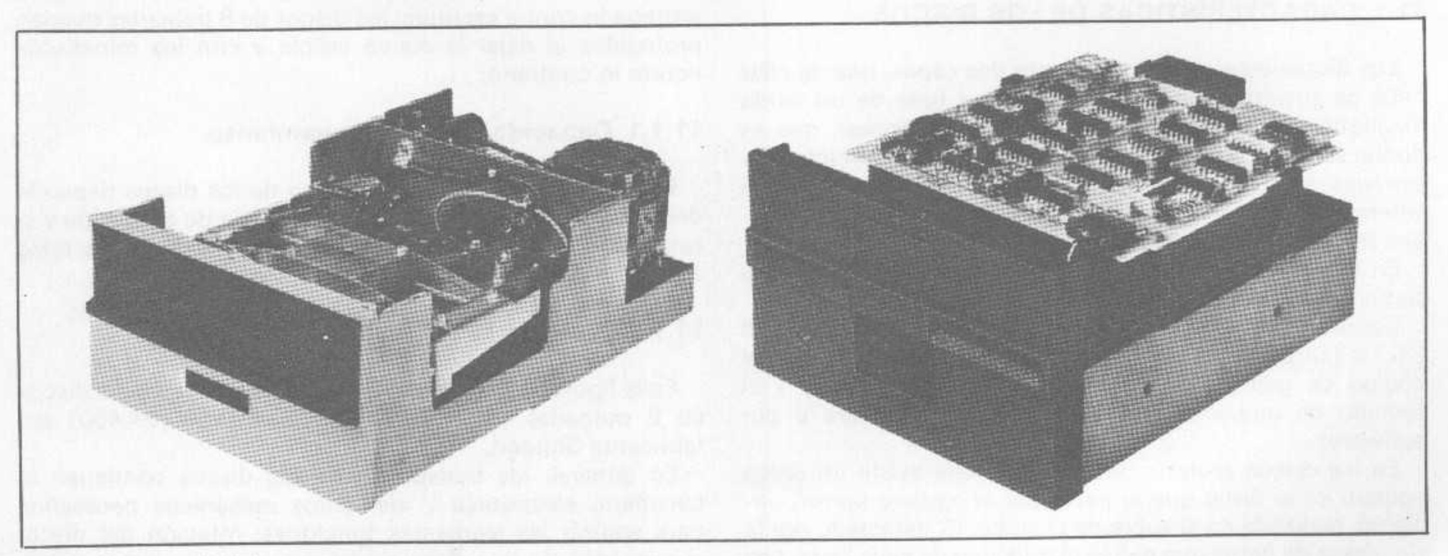


Figura 11.2 Fotografía de los transportes

11.1.2.6 *Disco preparado*

Es una señal que el transporte envía al controlador indicándole que hay un disco, que éste gira a una velocidad permitida y que las alimentaciones de continua del transporte son correctas.

11.1.2.7 *Disco protegido*

Como hemos comentado anteriormente, existe un mecanismo físico que nos permite proteger los discos contra escritura. Cuando el disco está protegido, el transporte bloquea la escritura y envía una señal al circuito controlador.

11.1.2.8 *Pista superior a la 43*

Debido a que en las pistas más altas, que corresponden a las de longitud más corta, la densidad de flujo magnético es mayor, conviene disminuir la corriente de escritura. Para ello existe en algunos transportes de discos un circuito que es disparado desde el controlador externo cuya misión es la de disminuir la corriente de grabación.

11.1.2.9 *Escritura-lectura de la información*

Corresponden a esta parte todos los mecanismos necesarios para grabar y recuperar la información sobre el material magnético. Los mecanismos empleados para ello en los discos flexibles son los mismos que los ya empleados en discos duros o cintas magnéticas. Podemos suponer el material magnético como formado por un gran número de partículas orientadas de manera aleatoria en el momento de la fabricación del disco. Cuando estas partículas son sometidas a un campo magnético suficientemente potente se orientan en la dirección provocada por el campo, de forma

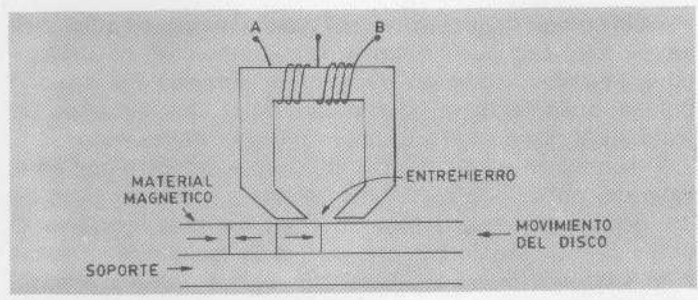


Figura 11.3 Cabeza de lectura-escritura

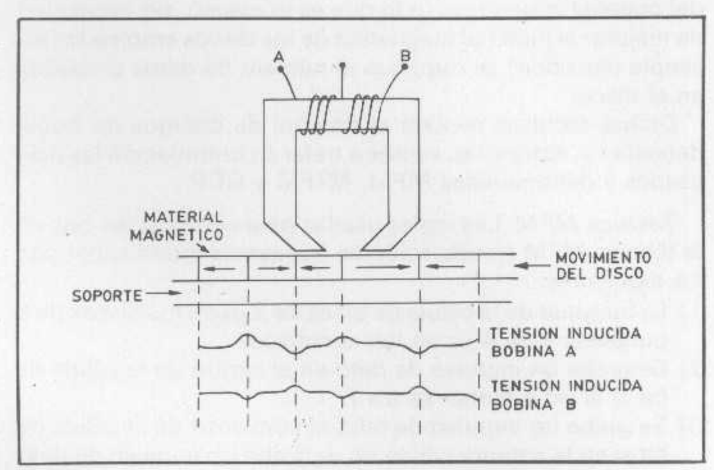


Figura 11.4 Proceso de lectura

Características	*Disco de 8 pulgadas*	*Disco de $5^1/_4$ pulg.*
Velocidad de rotación	360 vueltas/minuto	300
Tiempo de acceso entre pistas	3 a 10 milisegundos	5 a 40
Tiempo de estabilización de la cabeza	0 a 20 ms	15 a 20
Tiempo de carga de la cabeza	25 a 60 ms	35 a 75
Velocidad de transferencia	SD: 250 kilobits/s DD: 500 kilobits/s	125 250
MTBF	5000 a 8000 horas	Igual
MTFR	30 minutos	Igual
Vida media	5 años	Igual
Errores software	1 cada 10^8 bits leídos	Igual
Errores hardware	1 cada 10^{11} bits leídos	Igual
Errores de búsqueda	1 cada 10^6 búsquedas	Igual
Alimentación	Alterna 110-220 voltios 50-60 Hz Continua +24 ó +48 V +5, −5 ó +12 V	+5 y +12 voltios
Consumo	50 a 70 vatios	10 a 18 vatios
Tamaño	22 × 36 × 11 cm	14 × 20 × 8 cm

Tabla 11.2 Características de los transportes de los discos

que cuando éste desaparece quedan en esa posición. Para generar dicho campo magnético se emplea una cabeza consistente en un núcleo de ferrita al que se han arrollado un par de bobinas y realizado un corte que constituye el entrehierro (figura 11.3). Al pasar corriente por las bobinas se produce un campo magnético en el entrehierro de forma que las partículas magnéticas se orientan en la dirección de dicho campo. Un cambio en las corrientes de las bobinas hace que el sentido del campo magnético cambie y también el sentido de orientación de las partículas.

Para recuperar la información se emplea la misma cabeza, de forma que al pasar el disco sobre el entrehierro, los cambios de orientación de las partículas producen unos cambios de tensión en las bobinas A y B (figura 11.4).

En la tabla 11.2 damos los valores típicos de algunas características de los transportes de discos de 8 y 5 $^1/4$ pulgadas.

Representamos en la figura 11.5 los principales bloques mecánicos y eléctricos del SA800 así como las señales que usa para su comunicación con el controlador del transporte. Dicho transporte ha sido usado en los dos diseños prácticos realizados y descritos en el apartado 11.6.

Para mayor información del significado y forma de onda de las señales deberá leerse el manual del fabricante: SA800/801 Diskette Storage Drive. Shugart Associates.

11.2 CODIFICACION DE LA INFORMACION A GRABAR

La información lógica tratada por el computador está codificada en formato NRZ (No-return-zero). Dicho código o formato de bit no es útil para ser escrito y luego leído correctamente en el material magnético del disco. Así pues, es necesaria una codificación adicional de los datos a escribir en el disco y una decodificación de la información leída del disco a formato NRZ para que ésta pueda ser tratada por el microprocesador.

Los tipos de codificación empleados en los discos flexibles los podemos dividir en dos grupos: los denominados códigos de simple densidad y los de doble densidad. En

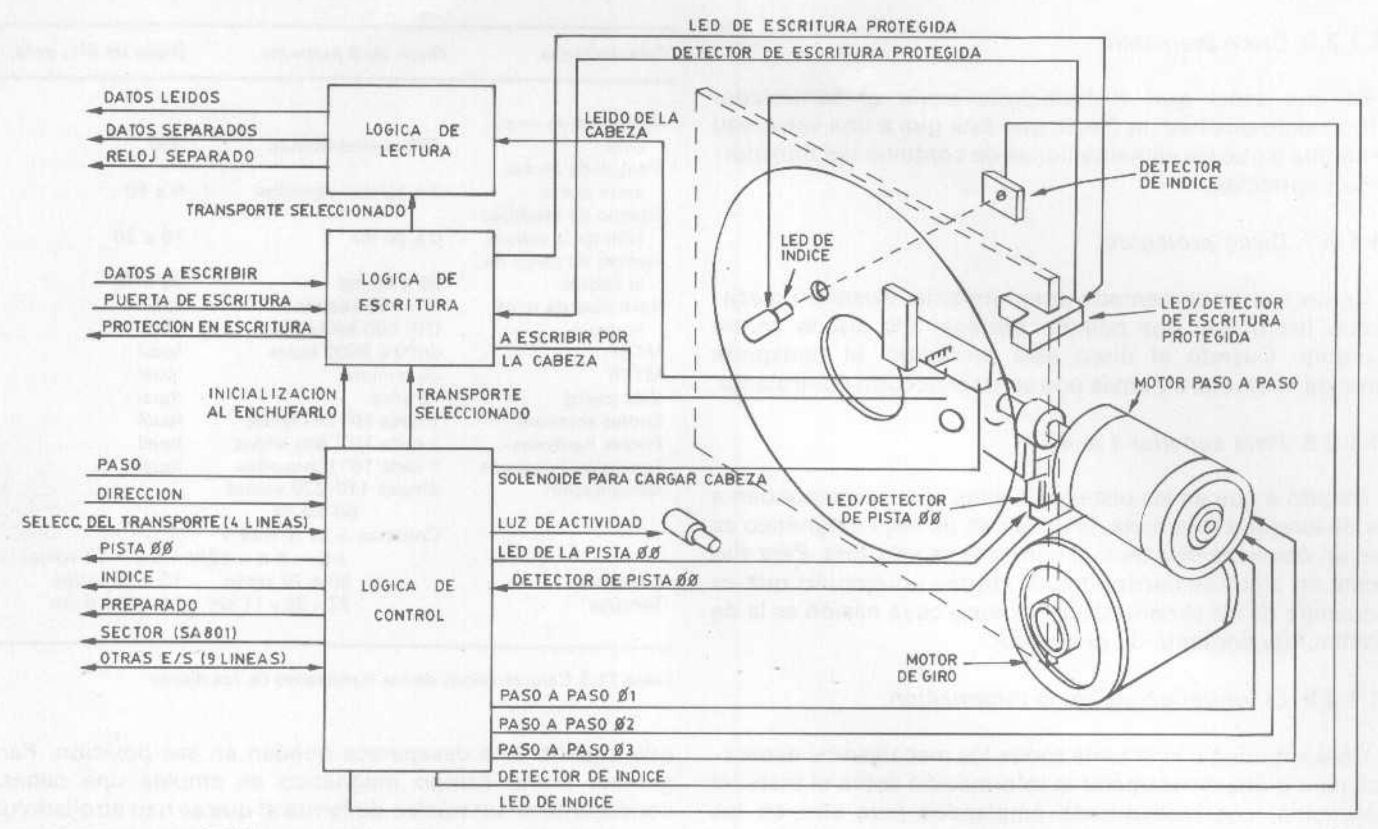


Figura 11.5 Transporte SA800

cualquiera de dichos códigos, el intervalo correspondiente a un bit en formato NRZ define la denominada célula de bit. La anchura de la célula de bit es de 4 μs en los discos de 8 pulgadas y de 8 μs en los minidiscos.

11.2.1 Códigos de simple densidad

El más usado es el denominado FM (modulación en frecuencia). En la figura 11.6 representamos la secuencia de bits 111010001, el formato de la señal que transmitiría una interfaz serie en NRZ, la secuencia equivalente en código FM y las longitudes y polarizaciones del medio magnético correspondientes a los bits anteriores.

Cada célula de bit contiene un impulso de reloj, *R*, al comienzo de ella y un impulso adicional de dato, en el centro de la misma si el bit correspondiente es un 1 y nada si el bit es 0.

La secuencia de datos y reloj es transformada en cambios de polarización, dentro del disco tal como representamos en la figura 11.3. La anchura de los impulsos enviados al

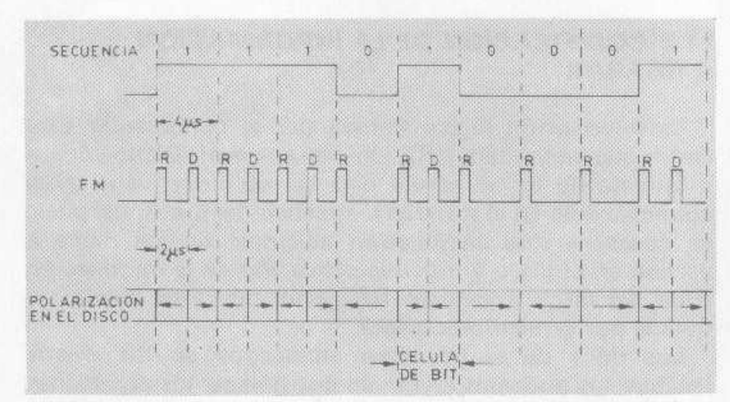


Figura 11.6 Código de grabación FM

transporte del disco por el controlador y la de los leídos varía entre los márgenes de 200 ns y 1 μs dependiendo del transporte usado.

11.2.2 Códigos de doble densidad

En el código de grabación FM usamos impulsos adicionales de reloj (*R*) que no contienen información de datos y cuya finalidad, como ya veremos al describir los diseños prácticos, es la de ayudar a sincronizar los circuitos de lectura del disco y recuperación posterior de los datos.

Sin embargo, es un alto precio a pagar debido a que en el caso de grabar una secuencia de bits a «1» el número de inversiones de polarización del material magnético es el doble del número de bits que contienen realmente información. Este hecho ha llevado consigo el que hayan aparecido técnicas de codificación de bits de forma que sin aumentar el número de cambios de flujo por unidad de longitud dentro del material magnético (o lo que es lo mismo, sin necesidad de mejorar el material magnético de los discos empleados en simple densidad) se duplique el número de datos grabados en el disco.

Dichas técnicas reciben el nombre de códigos de doble densidad y, entre ellas, vamos a tratar a continuación las más usadas y denominadas MFM, M^2FM y GCR.

Técnica MFM. Las reglas usadas para codificar los bits en la técnica MFM (modulación en frecuencia modificada) son las siguientes:

1) La longitud de la célula de bit es de 2 μs en los discos de 8 pulgadas y de 4 μs en los minidiscos.
2) Se graba un impulso de dato en el centro de la célula de bit si el bit a grabar es un 1.
3) Se graba un impulso de reloj al comienzo de la célula de bit si en la anterior célula no se grabó un impulso de dato y el bit a grabar en esta célula es un 0.

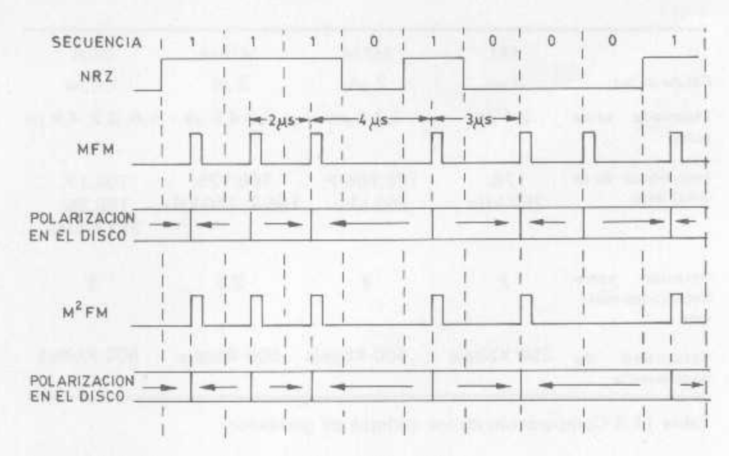


Figura 11.7 Códigos de grabación MFM y M^2FM

Técnica M^2FM. Las reglas usadas para codificar los bits en la técnica M^2FM (modulación en frecuencia modificada) son las siguientes:

1) La longitud de la célula de bit es de 2 μs en los discos de 8 pulgadas y de 4 μs en los minidiscos.
2) Se graba un impulso de dato en el centro de la célula de bit si el bit a grabar es un 1.
3) Se graba un impulso de reloj al comienzo de la célula de bit si ningún impulso de dato o reloj ha sido escrito en la célula de bit anterior y el bit a grabar en esta célula es un 0.

En la figura 11.7 representamos la misma secuencia de datos a grabar ya empleada en el apartado de FM, los modelos de las células de bit asociadas y los cambios de polarización de las partículas en el medio magnético del disco para los códigos de grabación MFM y M^2FM.

Técnica GCR. Esta técnica denominada GCR (grabación en grupos codificados) consiste en asociar a cada grupo de 4 bits de información un grupo de 5 bits de forma que no hayan nunca más de dos ceros consecutivos. La figura 11.8 representa dicha correspondencia.

Grupo a grabar	*Grupo grabado*
0000	11001
0001	11011
0010	10010
0011	10011
0100	11101
0101	10101
0110	10110
0111	10111
1000	11010
1001	01001
1010	01010
1011	01011
1100	11110
1101	01101
1110	01110
1111	01111

Figura 11.8 Código de grabación GCR

11.2.3 Comparación entre los códigos de grabación

Después de describir las reglas usadas para codificar los bits de NRZ a FM, MFM, M^2FM y GCR vamos a establecer algunas comparaciones entre los diferentes códigos. Las ventajas e inconvenientes se basarán en las dificultades encontradas al codificar la información para grabarla en el medio magnético; al decodificar la información leída del medio para convertirla en NRZ y en los problemas planteados por el propio medio magnético y las cabezas empleadas en la lectura-escritura.

El código FM tiene como ventajas el hecho de que solamente usa dos frecuencias y que la codificación y decodificación son muy sencillas, pudiéndose realizar esta última con monoestables. El impulso de reloj es necesario en la lectura para no perder la sincronización en el caso de que existan varios ceros seguidos. La desventaja más clara la constituye el hecho de que empleando un ancho de banda igual que los códigos de doble densidad solamente permite almacenar la mitad de información.

Un fenómeno que aparece en las cintas magnéticas y discos magnéticos es el denominado desplazamiento de bit. Dicho fenómeno es debido a las características del medio magnético y a la cabeza de lectura-escritura y consiste en que los campos magnéticos existentes cerca de los cambios de flujo al grabar impulsos inversos próximos hace que se interfieran de forma que la amplitud de la señal leída es menor que la nominal y los picos de dichas señales se encuentran también más separados que la distancia nominal. Como consecuencia de estos desplazamientos el circuito recuperador de los datos puede perder el sincronismo haciendo que los datos leídos no sean correctos.

Dicho fenómeno tiene el mismo valor absoluto usando códigos de simple o doble densidad. Sin embargo, en FM la anchura de la célula de bit es el doble que en los códigos de doble densidad, haciendo que el fenómeno de desplazamiento de bit no se deba tener en cuenta.

Para contrarrestar estos desplazamientos de los bits se debe usar una técnica denominada de precompensación. Dicha técnica se basa en el conocimiento que podemos tener a priori sobre la magnitud del desplazamiento de bit según la secuencia de datos a grabar. La precompensación consistirá en desplazar la escritura de un bit hacia la izquierda o hacia la derecha de forma que en la lectura los datos sean presentados al circuito separador en posiciones lo más cercanas posibles a su valor nominal.

En la figura 11.9 representamos la precompensación necesaria para las secuencias 011, 110, 0001 y 1000 que son las que necesitan precompensación en MFM.

En dicha figura P.B.N significa la posición que debería ocupar el bit normalmente en MFM y P.B.C es la posición que ocupa dicho bit después de la compensación. El desplazamiento de los bit precompensados, Δ, depende del

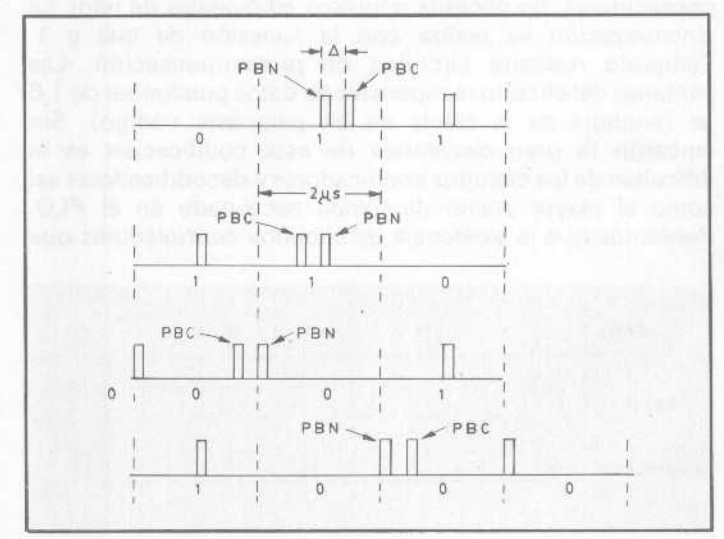


Figura 11.9 Precompensación en MFM

transporte y sus valores están comprendidos entre 100 y 300 ns. En el diseño de la sección 11.6 describiremos un circuito precompensador. M^2FM sólo necesita precompensar el bit segundo empezando por la izquierda en 011x, 110x y 0001.

Los códigos de doble densidad necesitan un circuito más complicado que los de simple densidad para la recuperación de los datos y convertirlos a NRZ. Normalmente este circuito está basado en un PLO (oscilador enganchado en fase). Durante el intervalo de cada célula de bit, dicho PLO debe abrir una ventana para el dato y otra para el reloj de forma que si, durante la ventana del dato se encuentra un impulso, éste es tomado como bit a 1 y si no lo encuentra, como bit a 0.

En la figura 11.10 se representan la secuencia de impulsos leídos del disco, las posibles ventanas generadas por un PLO y los datos recuperados.

En el diseño del controlador de disco para doble densidad (apartado 11.6) describiremos un circuito recuperador de datos.

El código M^2FM genera menos impulsos para grabar durante una secuencia de bits a 0, lo cual implica que la mayor parte de los desplazamientos de bit ocurrirán durante los impulsos correspondientes a los bits a 1. Esto lleva consigo que el PLO puede sincronizarse sólo con los bits a 0 ignorando los bits a 1 lo cual constituye una pequeña ventaja. Sin embargo, la sincronización cuando no haya bits a 0 o haya pocos será más difícil.

Puesto que la mínima distancia entre los bits de reloj en M^2FM es mayor que la mínima distancia entre los bits de datos, el desplazamiento de bits correspondiente a los bits de reloj será menor que el correspondiente a los bits de datos de forma que se podrá emplear un PLO con ventanas asimétricas, pudiendo ser éstas de 1,2 μs para los bits de datos y 0,8 μs para los de reloj lo cual aumenta considerablemente la fiabilidad en la recuperación de los datos.

Sin embargo, los PLO necesitados en MFM son más sencillos de realizar ya que este código emplea menos frecuencias diferentes. Otro parámetro adicional en favor de MFM es el hecho de que ha sido estandarizado por IBM lo cual ha hecho que la mayoría de los controladores integrados lo soporten, no ocurriendo lo mismo con el código M^2FM.

El código GCR evita la pérdida de sincronización en el circuito recuperador de datos, al no permitir la existencia de más de dos ceros consecutivos. Al transformar grupos de 4 bits de grupos de 5 y no permitir más de dos ceros consecutivos, no necesita impulsos adicionales de reloj. La sincronización se realiza con la sucesión de bits a 1. Tampoco necesita circuitos de precompensación. Las ventanas del circuito recuperador de datos pueden ser de 1,6 μs (anchura de la célula de bit para este código). Sin embargo la gran desventaja de esta codificación es la dificultad de los circuitos codificadores y decodificadores así como el mayor ancho de banda necesitado en el PLO. Pensamos que la existencia de circuitos controladores que integraran en una sola pastilla todas las funciones requeridas para la codificación y decodificación en código GCR harían de éste, debido a su mayor fiabilidad, uno de los candidatos a ser empleados en el diseño de los subsistemas de discos flexibles.

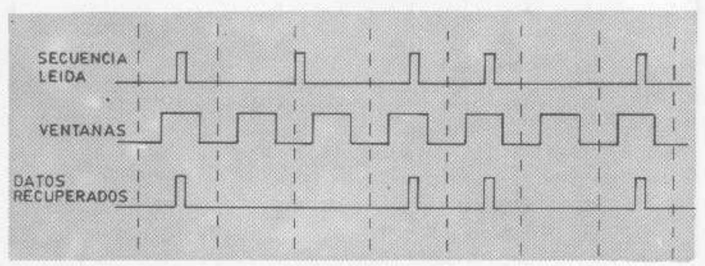


Figura 11.10 Recuperación de datos en MFM

	FM	MFM	M^2FM	GCR
Célula de bit	4 μs	2 μs	2 μs	1,6 μs
Espaciado entre pulso	2,4 μs	2,3,4 μs	2,3,4,5 μs	1,6; 3,2; 4,8 μs
Frecuencias de la señal leída	125; 250 kHz	125;166,7; 250 kHz	100;125; 166,7; 250 kHz	104,17; 156,25; 312,5 kHz
Relación entre frecuencias máximas	2	2	2,5	3
Velocidad de transferencia	250 Kbits/s	500 Kbits/s	500 Kbits/s	500 Kbits/s

Tabla 11.3 Comparación de los códigos de grabación

En la tabla 11.3 damos unos valores comparativos para los diferentes códigos de bit.

11.3 FORMATOS DE PISTA

Entendemos por formato de pista la manera de colocar la información dentro de la pista. Básicamente, cada pista debe contener tres tipos de información: la que pertenece propiamente a los datos del usuario, la que identifica a la anterior y la necesaria para la sincronización de los circuitos mecánicos y de lectura-escritura del usuario.

Aunque cada usuario podría definir su propio formato, existen algunos ya normalizados que vamos a describir a continuación. En concreto, trataremos con los formatos IBM 3740 (para simple densidad), IBM 34 (para doble densidad) un posible formato a usar en los minidiscos y un formato para discos sectorizados por hardware y usado por Zilog en sus sistemas de desarrollo.

11.3.1 IBM 3740

Este formato es el más usado en la actualidad para discos sectorizados por software y codificación en FM. En la figura 11.11 representamos dicho formato.

Cada pista comienza cuando se detecta el impulso índice. Como vemos existen unos campos denominados «gap», usados para separar los campos de información de la pista y que sirven para compensar las tolerancias mecánicas y eléctricas en la posición de la cabeza de lectura-escritura y las pequeñas variaciones rotacionales del disco de forma que los discos pueden ser leídos sobre diferentes transportes y

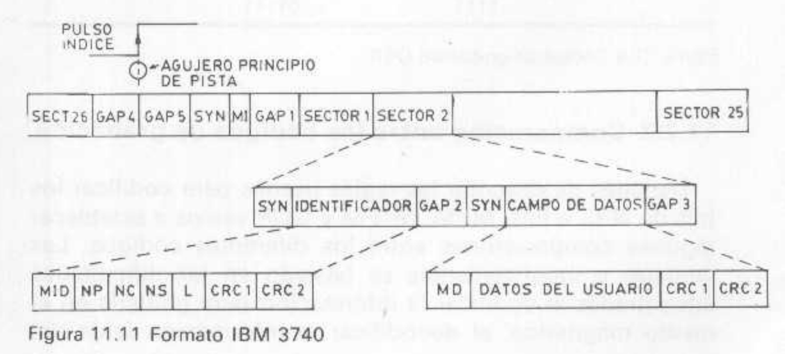


Figura 11.11 Formato IBM 3740

Marca	Reloj	Datos
Indice (MI)	D7	FC
Identificador (MID)	C7	FE
Datos (MD)	C7	FB
Datos borrados (MDB)	C7	FB

Tabla 11.4 Composición de las marcas

sistemas. Además también permiten la actualización de un campo de información sin afectar a los campos adyacentes ya que el inicio de cada campo puede variar ligeramente.

Inmediatamente antes de cada campo de información existe un campo que hemos denominado SYN (sincronización) y cuya misión es la de sincronizar correctamente el circuito separador de datos para que éstos puedan recuperarse.

Por otra parte, en el formato expresado en la figura 11.11 existen 26 sectores. Cada sector está compuesto por un campo SYN, un campo identificador del sector, el gap 2, otro campo SYN, el campo de datos y el gap 3.

El campo identificador contiene información que, como su propio nombre indica, sirve para identificar al sector. A su vez, está compuesto por 7 octetos que representan respectivamente la marca del identificador (MID), el número de pista sobre la que nos encontramos (NP), el número de la cara para discos de doble cara (NC), el número del sector (NS), información para calcular la longitud del sector (L) y los octetos de detección de errores (CRC1, CRC2). El valor de los dos últimos octetos es el resultado de dividir los 5 octetos anteriores por el polinomio $x^{16}+x^{12}+x^{5}+1$ después de poner el valor del registro de CRC con todo unos antes de comenzar la división.

El campo de datos está formado por un octeto que contiene la marca de datos (MD), 128 octetos de datos del usuario y 2 octetos con el CRC de los 129 anteriores.

Existen en el disco 4 tipos de octetos especiales, denominados marcas. El primero de ellos se denomina marca del índice (MI) y está colocado al principio de cada pista. El segundo corresponde al primer octeto del campo identificador del sector y se denomina marca del identificador (MID). El tercero y cuarto son dos variaciones de la marca de datos (MD), que es el primer octeto del campo de datos. Dicho octeto puede contener una marca de datos (MD), o una marca de datos borrados (MDB). Dichas marcas están formadas por un conjunto de 8 células de bit no repetidas en ninguna otra parte del disco. Para ello se graban impulsos de reloj y de datos de forma diferente a como se graban todas las otras informaciones. En concreto, la tabla 11.4 recoge los modelos de dichas marcas.

Así como la misión de los campos SYN es la de sincronizar el circuito de lectura a nivel de bit, es decir para poder muestrear correctamente cada bit, la misión de las marcas es la de sincronizar el circuito que ensamble los bits y los convierta en octetos para presentarlos correctamente al microprocesador.

Como última parte de este apartado, representamos en la tabla 11.5 las diferentes longitudes de los campos para el formato IBM 3740.

11.3.2 IBM 34

El formato de pista IBM 34 se emplea para codificación en doble densidad MFM. Básicamente el formato es muy parecido al IBM 3740 y sólo se diferencia en el tamaño de los campos de datos y gaps por una parte y en los códigos de las marcas y algunos nuevos campos de sincronización. En la tabla 11.6 representamos los diferentes campos dentro de una pista.

Como vemos, existen tres tipos diferentes de campos de sincronización además de los que existían en el IBM 3740: los tres octetos de sincronización de la marca índice, los tres octetos de sincronización del campo identificador de sector y los tres octetos de sincronización del campo de datos del sector.

11.3.3 Formatos para minidisco

Los formatos empleados para minidiscos sectorizados por software son muy parecidos a los ya descritos para los discos de 8 pulgadas.

Como ejemplo vamos a describir uno que propone Shugart. Las diferencias existentes al compararlo con el IBM 3740 (figura 11.11) son los siguientes:

— Posee 18 sectores de datos.
— El gap 5 tiene una longitud de 16 octetos a FF.
— Los campos SYN son 4 octetos a 00.
— No existe marca del índice (MI).
— No existe el gap 1.
— El campo identificador está compuesto de 5 octetos: marca identificadora (MID), número de pista (NP), número de sector (NS) y dos octetos de CRC.

Formato \ Campo	GAP5	SYNC	MI	GAP1	SYN	Identificador	GAP2	SYNC	Campo de datos	GAP3	GAP4
26 sectores 128 octetos por sector	40 octetos A FF	6 octetos A 00	1 octeto A FC Y reloj A C7	26 octetos A FF	6 octetos A 00	7 octetos	11 octetos A FF	6 octetos A 00	131 octetos	27 octetos A FF	247 octetos A FF
15 sectores 256 octetos por sector									259 octetos	42 octetos A FF	170 octetos A FF
8 sectores 512 octetos por sector									515 octetos	311 octetos A FF	58 octetos A FF

← Repetido 26.15 u 8 veces (una por cada sector) → (SYN … GAP3)

← Aproximadamente 5208 octetos/pista →

Tabla 11.5 Formato IBM 3740

Campo / Formato	GAP5	SYN	octet de SYN de pista	MI	GAP1	SYN	octet. de SYN identif.	identificador	GAP2	SYN	octet. de SYN de sector	campo de datos	GAP3	GAP4
26 sector. 256 octet. por sector	80 octet. A 4E	12 octet. A00	3 octet. A C2	1 octet. A FC	50 octet. A 4E	12 octet. A 00	3 octet. A A1	7 octet.	22 octet. A 4E	12 octet. A 00	3 octet. A A1	259 octet	54 octet. A 4E	598 octet. A 4E
15 sect. 512 octet. por sector												515 octet	84 octet. A 4 E	400 octet. A 4E
8 sect. 1024 octet. por sect.												1027 octet.	116 octet. A 4E	654 octet. A 4E

Repetido 26,15 u 8 veces por cada sector

Aproximadamente 10416 octetos/pista

Tabla 11.6 Formato IBM 34

— El gap 2 tiene una longitud de 6 octetos a FF.
— El campo de datos es idéntico al IBM 3740.
— El gap 3 tiene 17 octetos a FF.
— El gap 4 tiene una longitud aproximada de 103 octetos a FF.

11.3.4 Formatos hardware

Para este tipo de discos no existen formatos normalizados tal vez debido a que han perdido mucho interés con la salida al mercado de circuitos controladores de los formatos IBM anteriormente expuestos.

Como ejemplo práctico vamos a describir el formato que usa Zilog en sus equipos de desarrollo con discos de 8 pulgadas. Cada pista tiene 32 sectores y el formato de cada uno de ellos es el expresado en la figura 11.12. Como vemos, después de detectar la marca de principio de sector viene un preámbulo de 16 octetos a 00 para sincronizar el circuito de lectura; inmediatamente después vienen los octetos DS y DP que indican el número de sector y número de pista. El campo de datos está formado por 128 octetos de datos seguidos de 4 octetos que contienen información para enlazar los sectores dentro de un fichero; esta información es usada por la parte del sistema operativo que maneja el disco. Los dos octetos que siguen contienen el CRC de los 134 octetos anteriores, y después de este campo viene el postámbulo que son octetos a 00. Este campo dura hasta llegar al agujero que indica el principio del siguiente sector.

11.4 INTERFAZ MICROPROCESADOR-TRANSPORTE DEL DISCO

En este apartado describiremos las funciones que deben ser realizadas por la interfaz que une el bus del sistema microcomputador empleado con el transporte físico del disco. Dichas funciones las clasificaremos en dos apartados: las relativas a la transferencia de datos y las que se encargan de control mecánico y eléctrico del transporte.

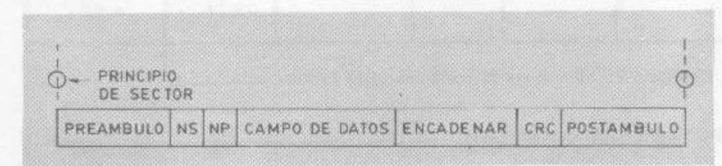


Figura 11.12 Formato para discos sectorizados por hardware

11.4.1 Transferencia de datos

Por una parte conocemos los mecanismos de entrada-salida de datos del sistema microprocesador y por otra hemos descrito los formatos de pista y codificaciones a nivel de bit empleados para grabar y recuperar los datos en el disco. Así pues, se deberán realizar las siguientes funciones:

1. Para la escritura en el disco, serialización de los datos entregados por el microprocesador, generación de la señal compuesta con arreglo al código de bit empleado, precompensación de esos datos si es necesario y generación de los códigos CRC para detección de errores.
2. Para la lectura del disco, separación de los datos a partir de la señal compuesta para obtener el sincronismo a nivel de bit, detección de las marcas especiales para obtener el sincronismo a nivel de octeto y comprobación de posibles errores de CRC.
3. En ambas operaciones deberá existir circuitería de sincronización entre la interfaz y el sistema microcomputador, así como los mecanismos necesarios para la transferencia de datos, control y estado.

11.4.2 Control mecánico y eléctrico del transporte

Dentro de este apartado se deberán realizar las siguientes funciones:

1. En relación al control de la cabeza de lectura-escritura se deberán generar las señales necesarias para moverla por entre las pistas del disco y para posicionarla sobre el material magnético para poder realizar las operaciones de lectura-escritura.
2. Se deberán detectar las señales enviadas por el transporte correspondiente a la situación de la cabeza sobre la pista 00 así como al paso de los agujeros correspondientes a los principios de pista y sector.
3. Detección de las señales enviadas por el transporte relativas a situaciones tales como disco protegido contra escritura o disco preparado para ser leído o escrito.
4. Se deberán enviar al transporte señales para permitir la escritura y seleccionar la cara correspondiente en los transportes de doble cara.

11.5 CONTROLADORES INTEGRADOS

Debido por una parte a la complejidad de las funciones

necesarias para realizar la interfaz entre el microcomputador y el transporte del disco y por otra a la enorme aceptación y uso de los discos flexibles, los fabricantes de microprocesadores han integrado gran parte de dichas funciones en una sola pastilla de forma que el precio de la interfaz ha disminuido considerablemente (alrededor de 10.000 pesetas) y la fiabilidad ha aumentado.

Como hemos dicho anteriormente, el número de funciones implantadas dentro del controlador ha aumentado desde los primeros controladores y es fácil pensar que en el futuro se realizarán verdaderos procesadores de entrada-salida dedicados al control específico de los transportes de disco.

A continuación vamos a describir brevemente los siguientes controladores integrados: WD1771, WD1781, WD1791, I8271, M6843 y PD765C. Dedicaremos especial atención a los controladores WD1771 y WD1791 que han sido los empleados en las realizaciones prácticas que describimos en el apartado 11.6.

11.5.1 Controlador WD1771

Es un controlador de la Western Digital fabricado en tecnología MOS de canal N y puerta de silicio con todas entradas y salidas compatibles TTL.

11.5.1.1 *Señales del controlador*

En la figura 11.13 representamos la posición relativa y nombre de las señales del controlador. Para comentar su significado las clasificamos en tres grupos: señales para el diálogo con el sistema microcomputador, señales para el diálogo con el transporte de los discos y alimentaciones.

11.5.1.1.1 Señales para el diálogo con el bus del μC

— DAL0-DAL7 (datos de entrada-salida). Constituyen el bus de comunicación de datos, control y estado entre el microcomputador y el controlador.
— $\overline{CS}$ (selección del controlador). Un nivel bajo en esta señal selecciona al controlador permitiendo su comunicación con el microcomputador.
— A_0-A_1 (líneas de selección de registros). Seleccionan el registro interno sobre el que se va a realizar la operación.
— $\overline{RE}$ (lectura). Si esta señal está a nivel bajo se realiza una lectura del registro indicado por A_1-A_0.
— $\overline{WE}$ (escritura). Si esta señal está a nivel bajo se realiza una escritura del registro indicado por A_1-A_0.
— CLK (reloj). El controlador necesita para generar sus tiempos internos un reloj de 2 MHz ±1 % y onda cuadrada.

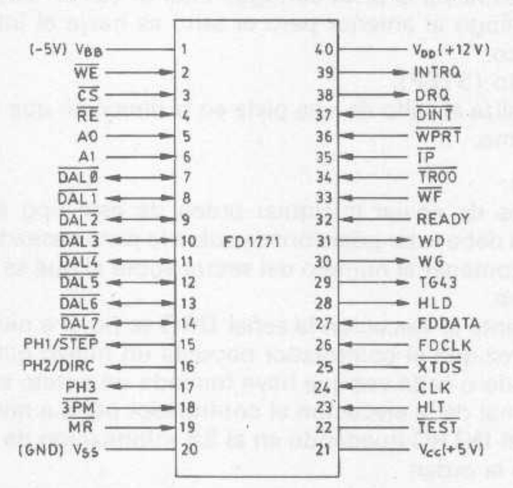


Figura 11.13 Señales del WD1771

— DRQ (sincronización de datos). DRQ e INTRQ constituyen las dos señales de sincronización entre el controlador y el microcomputador. DRQ se pone a nivel alto siempre que el controlador ha ensamblado un nuevo octeto en lectura o necesita un nuevo octeto en escritura.
— INTRQ (sincronización de órdenes). Esta señal se pone a nivel alto para indicar al microcomputador que el controlador ha terminado de ejecutar una orden. El resultado de dicha orden queda reflejado en un registro de estado perteneciente al controlador.
— $\overline{MR}$ (reinicialización). Un nivel bajo en esta señal de entrada pone a cero el registro de órdenes, y posiciona la cabeza sobre la pista 00.

11.5.1.1.2 Señales para el diálogo con el transporte

— WD (datos a escribir). Esta señal contiene la señal compuesta (datos + reloj) que el controlador ha formado para ser grabada en el disco. La duración de los impulsos es de 500 ns.
— $\overline{XTDS}$ (separación de datos externa). Un nivel a cero en esta entrada al controlador le indica que se ha realizado externamente la separación de datos y reloj de la señal compuesta leída por el transporte. Un nivel a uno le indica que él deberá separar internamente los datos y el reloj.
— FDDATA (datos leídos). Por esta entrada recibe el controlador la señal compuesta leída del disco, si no han sido separados los datos y reloj, o la señal de datos si éstos han sido separados.
— FDCLOCK (reloj de los datos). Por esta entrada recibe el controlador la señal de reloj, si ya ha sido obtenida por el circuito separador externo. Cuando la separación de datos deba ser realizada por el controlador (XTDS = 1), la señal FDCLOCK deberá polarizarse a 5 voltios.
— HLD (cargar cabeza). Esta señal es enviada hacia el transporte del disco para que éste baje la cabeza sobre la superficie del disco.
— HLT (cabeza cargada). Después de ordenar bajar la cabeza, el controlador debe esperar a que ésta se encuentre sobre la superficie magnética. Este tiempo depende del transporte empleado. HLT a nivel uno le indica al controlador que la cabeza se encuentra ya sobre el disco.
— PH1/STEP, PH2/DIR, $\overline{PH3}$ y 3PM. Estas cuatro señales son las empleadas por el controlador para gobernar el motor paso a paso que desplaza la cabeza de lectura-escritura sobre las diferentes pistas. Si 3PM se polariza a uno, las señales PH1, PH2 y $\overline{PH3}$ sirven para gobernar motores de 3 fases. Si 3PM se pone a nivel cero, PH1/STEP y PH2/DIR sirven para gobernar los motores de 2 fases.
— TG43 (pistas mayor que 43). Es una señal que envía el controlador al transporte siempre que quiera realizar una escritura en una pista superior a la 43. Los transportes que admiten esta señal disminuyen la corriente de escritura.
— WG (permitir escritura). Esta señal es enviada al transporte para que éste permita los circuitos de escritura.
— READY (disco preparado). Señal de entrada al controlador indicándole si puede realizar operaciones de lectura-escritura.
— WF (fallo de escritura). Esta entrada al controlador le indica que existe un fallo de escritura.

— $\overline{TR00}$ (pista cero). Esta señal indica al controlador que la cabeza de lectura-escritura está sobre la pista 00.
— $\overline{IP}$ (pulso del índice). Señala al controlador la detección del agujero principio de pista.
— $\overline{WPRT}$ (escritura protegida). Esta entrada le indica al controlador que el disco que tiene el transporte en ese momento está protegido contra escritura.
— $\overline{DINT}$ (inicialización del disco). Si esta entrada está a nivel cero, le indica al controlador que no puede realizar la orden de escribir una pista.
— $\overline{TEST}$ (comprobación). Esta entrada es usada para comprobar el funcionamiento interno del controlador.

11.5.1.1.4 Alimentaciones

— V_{BB}, V_{CC}, V_{DD} y V_{SS}. Las tensiones de estas entradas deberán ser −5 V, +5 V, +12 V y tierra (masa) respectivamente.

11.5.1.2 *Diagrama de bloques del controlador*

El diagrama de bloques del controlador está representado en la figura 11.14.

Los registros internos que pueden ser accedidos por el programador son los siguientes:

— Registro de datos (DR).
Es empleado para la transferencia de datos en las operaciones de lectura-escritura. Debe ser cargado con el valor de la pista en la que queremos posicionar la cabeza de lectura-escritura antes de ejecutar la orden de búsqueda de pista (apartado 11.6.)
— Registro de pista (TR).
Los 8 bits de este registro contienen el valor de la pista donde se encuentra la cabeza de lectura-escritura. Su contenido puede ser leído y escrito y se actualiza cada vez que la cabeza salta de una pista a otra.
— Registro de sector (SR).

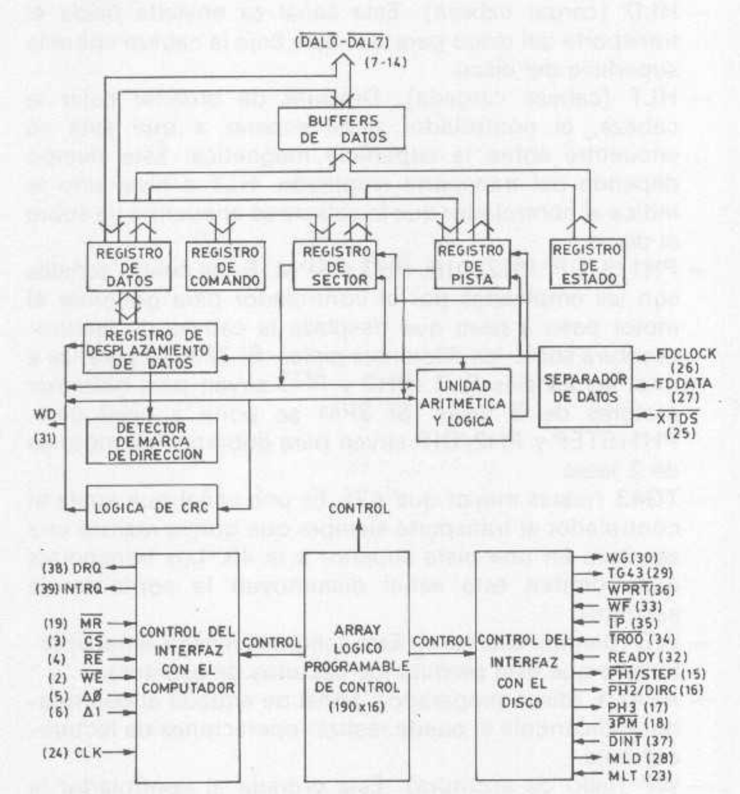


Figura 11.14 Diagrama de bloques del WD1771

$A1$	$A0$	$\overline{RE}$	$\overline{WE}$
0	0	STR	CR
0	1	TR	TR
1	0	SR	SR
1	1	DR	DR

Tabla 11.7 Selección de registros en el WD1771

Contiene el número del sector que se quiere leer o escribir. Su contenido puede ser leído o escrito bajo programa.
— Registro de órdenes (CR).
Su contenido corresponde al código de la orden que se está ejecutando. Salvo para la orden de interrupción forzada, dicho registro no se puede volver a escribir mientras el controlador está ejecutando la orden anterior.
— Registro de estado (STR).
Contiene información acerca del estado del controlador.

La selección de estos registros se realiza como indicamos en la tabla 11.7.

Existen otros registros no accesibles al programador tales como los registros de desplazamiento y generadores de CRC.

11.5.1.3 *Ordenes ejecutadas por el controlador*

El controlador es capaz de ejecutar 11 órdenes diferentes que dividimos por su similitud en 4 grupos:

Tipo I:

Las órdenes de este tipo no necesitan transferencias de datos durante su ejecución. Al final de ésta, el controlador genera la señal INTRQ y el SR contiene la información relativa a cómo se realizó la orden.

— Ir a la pista 00 (Restore).
Al recibir esta orden, si la señal $\overline{TR00}$ no está a nivel bajo, el controlador genera la secuencia necesaria en las señales de control del motor paso a paso hasta posicionar la cabeza sobre la pista 00.
— Búsqueda de una pista (SEEK).
Al recibir esta orden, el controlador genera la secuencia de señales necesarias para posicionar la cabeza sobre el número de pista contenido en el DR.
— Salto hacia la pista contigua exterior (STEP-OUT).
Al recibir esta orden el controlador genera la secuencia de señales necesarias para poner la cabeza sobre la pista contigua exterior de la que se encontraba anteriormente.
— Salto hacia la pista contigua interior (STEP-IN).
Análogo al anterior pero el salto es hacia el interior del disco.
— Salto (STEP).
Realiza el salto de una pista en la dirección que realizó el último.

Tipo II:

Antes de enviar cualquier orden de este tipo al CR, la cabeza debe estar posicionada sobre la pista deseada y el SR debe contener el número del sector sobre el que se realizará la orden.

Durante la ejecución la señal DRQ se pone a nivel activo cada vez que el controlador necesita un nuevo octeto para escribirlo o cada vez que haya formado un octeto el lectura.

Al final de la ejecución el controlador pone a nivel activo la señal INTRQ quedando en el SR información de cómo se realizó la orden.

— Leer sector.

Después de recibir esta orden el controlador genera las señales correspondientes para bajar la cabeza y cuando detecta el sector deseado comienza a leer su contenido presentando cada octeto al microcomputador por medio de la señal DRQ.

— Escribir sector.

Después de recibir esta orden, el controlador genera las señales correspondientes para bajar la cabeza. Cuando detecta el sector deseado comienza la secuencia de petición de datos al microprocesador y el envío de la señal compuesta al transporte.

Ambas órdenes permiten realizar las operaciones de lectura-escritura sobre varios sectores consecutivos hasta que se detecte la señal de comienzo de pista o se envíe al controlador una orden de interrupción forzada.

Tipo III:

La sincronización con el microprocesador es la misma que la necesitada por las órdenes de tipo II.

— Leer pista.

Después de recibir esta orden, el controlador ordena bajar la cabeza y después de encontrar la marca índice comienza a leer todos los octetos de la pista (incluidos los gaps) y los presenta al microprocesador a través de la señal DRQ. La orden acaba al detectar de nuevo la señal MI.

— Escribir pista.

Análogo al anterior y sirve para grabar toda una pista. Una de las mayores aplicaciones de esta orden es la de formatear un disco virgen.

— Leer campo identificador.

Después de recibir esta orden, el controlador ordena bajar la cabeza y lee el primer campo identificador encontrado. Los 7 octetos correspondientes a ese campo son enviados hacia el microprocesador.

Tipo IV:

En este tipo de órdenes sólo se encuentra la de interrupción forzada. Esta orden es empleada para acabar bruscamente con la orden que se está ejecutando.

11.5.1.4 *Código de las órdenes*

En la tabla 11.8 representamos el código de las diferentes órdenes ejcutadas por el FD1771.

Tipo	*Orden*	$b_7b_6b_5b_4b_3b_2b_1b_0$
I	Ir a pista 00 (restore)	0 0 0 0 h v r_1 r_0
	búsqueda de pista (see)	0 0 0 1 h v r_1 r_0
	Salto (Step)	0 0 1 u h v r_1 r_0
	Salto pista contigua interior (step-in)	0 1 0 u h v r_1 r_0
	Salto pista contigua exterior (step-out)	0 1 1 u h v r_1 r_0
II	Leer sector(es) (read command)	1 0 0 m b E 0 0
	Escribir sector(es) (write command)	1 0 1 m b E a_1 a_0
III	Leer campo identificador (read address)	1 1 0 0 0 1 0 0
	Leer pista (read track)	1 1 1 0 0 1 0 S
	Escribir pista (write track)	1 1 1 1 0 1 0 0
IV	Interrupción forzada (force interrupt)	1 1 0 1 I_3 I_2 I_1 I_0

Tabla 11.8 Código de las órdenes del WD1771

r_1 r_0	*CLK=2 MHz test=1*	*CLK=1 MHz test=1*	*CLK=2 MHz test=0*	*CLK=1 MHz test=0*
0 0	6 ms	12 ms	400 μs	800 μs
0 1	6 ms	12 ms	400 μs	800 μs
1 0	10 ms	20 ms	400 μs	800 μs
1 1	20 ms	40 ms	400 μs	800 μs

Tabla 11.9 Velocidad de salto entre pistas (WD1771)

— r_1, r_0 sirven para indicar al controlador las velocidades del motor paso a paso al saltar de pista a pista. En la tabla 11.9 se dan estas velocidades.

— $h=1$ indica que la cabeza debe ser cargada al principio del mandato. Una vez enganchada la cabeza continúa así mientras recibe órdenes que indican enganche de cabeza o hasta que se hacen 2 revoluciones en el disco después de recibir la última orden que indicaba dicho enganche.

— El bit *V* indica, si está a 1, que debe realizarse una verificación sobre la última pista. Durante la verificación y después de generar los tiempos programados para HLT, es leído el primer campo identificador encontrado. La dirección de la pista del campo es comparada con el contenido del TR. Si son iguales y el CRC es correcto, se genera una interrupción y la orden acaba. En caso contrario, si el controlador no puede encontrar un sector con número de pista y CRC correcto después de 2 revoluciones del disco, finaliza la orden mandando una interrupción al microprocesador e indicando en el SR que la orden no pudo ser realizada correctamente.

— El bit *u* a 1 indica que el TR debe ser actualizado a cada salto realizado.

— Si el bit *E* lo ponemos a 1 le indicamos al controlador que debe generar la secuencia para bajar la cabeza sobre el disco.

— El bit *b*, junto con el contenido del octeto longitud del sector (octeto 5 del campo identificador del sector) determinan el número de octetos del campo de datos. Para formato IBM 3740, el bit *b* deberá ser igual a 1 de forma que el número de octetos de datos dentro de cada sector será de 128×2^n (n = 0, 1, 2 ó 3. Siendo *n* el contenido del octeto 5 anterior). Cuando *b* es igual a 0, el número de octetos viene determinado por el contenido del octeto 5 anterior multiplicado por 16 (n = 1, 2,...FF,00 correspondiente a 16, 32,... 4096 octetos).

— El bit *m* indica si se van a leer o escribir uno ($m=0$) o varios sectores ($m=1$). Si *m* es igual a 1 se van leyendo o escribiendo sectores hasta que el contenido de SR excede el valor 26 o se recibe una orden de interrupción forzada.

— Los bits a_1, a_0 de la orden de escritura sirven para indicarle al controlador qué tipo de marca de datos deberá escribir de arreglo con la tabla 11.10.

— El bit *S* de la orden leer pista sirve para indicarle al controlador si debe sincronizarse al encontrar cualquier marca de direcciones ($S=0$) o no ($S=1$).

— Los bits I_0-I_3 de la orden de interrupción forzada sirven

a_1 a_0	*Octeto de datos*	*Reloj*
0 0	FB	C7
0 1	FA	C7
1 0	F9	C7
1 1	F8	C7

Tabla 11.10 Tipos de datos encontrados

para indicarle al controlador bajo qué condiciones deberá generar una interrupción hacia el sistema microcomputador. En general se puede especificar más de una condición y éstas son:

I_0 = 1. Transición de disco no preparado a preparado
I_1 = 1. Transición de disco preparado a no preparado
I_2 = 1. A cada pulso índice
I_3 = 1. Inmediatamente

Si $I_0 - I_3$ son cero, la orden se termina inmediatamente y no se genera interrupción.

11.5.1.5 *Estado del controlador*

El registro de estado del controlador, SR, contiene información relativa a cómo se ejecutó la última orden de forma que cada vez que el controlador envíe una interrupción al sistema microcomputador, indicativa de que terminó de ejecutar una orden, se deberá decodificar el SR para saber cómo se ejecutó.

La tabla 11.11 contiene el significado de cada uno de los bits del SR en función de la orden ejecutada.

11.5.2 Controlador WD1781

Constituye una primera modificación del WD1771 para admitir códigos de grabación en doble densidad. Las diferencias funcionales que existen entre ellos son las siguientes:

— El WD1781 puede trabajar con simple y doble densidad.
— El WD1771 posee un circuito separador de datos y reloj interno, mientras que el WD1781 no lo posee.
— El WD1771 puede generar (en escritura) y detectar (en lectura) las marcas de direcciones de índice, sector y datos, mientras que el WD1781 no detecta ninguna de ellas y sólo las genera cuando trabaja en simple densidad.
— El WD1781 solamente puede controlar motores paso a paso de dos fases de forma que de las cuatro patillas necesitadas por el WD1771, libera dos (PH3 y $\overline{3PM}$) para realizar las funciones que describiremos más adelante.
— El WD1781 no necesita la alimentación de —5 voltios.

Como vemos, las diferencias entre el WD1771 y el WD1781 están en las señales de la interfaz con el transporte del disco y no en las señales de conexión al bus del

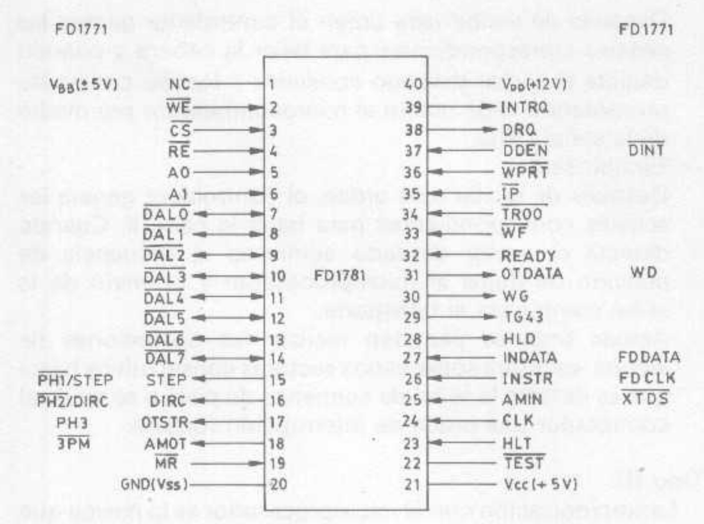


Figura 11.15 Señales del WD1781

microcomputador. De hecho el sistema microcomputador ve a ambos circuitos como si fueran el mismo en cuanto a señales eléctricas y órdenes que son capaces de ejecutar.

11.5.2.1 *Señales de controlador*

Solamente describiremos el significado de aquellas señales que son diferentes a las correspondientes en el WD1771. Para ello, en la figura 11.15 representamos un dibujo con la señales del WD1781 y las correspondientes del WD1771 en el caso de que sean diferentes.

— La patilla 1, correspondiente a la alimentación —5 voltios en el WD1771, no es usada por el WD1781 de forma que se dejará sin conectar.
— Las patillas 15 y 16 son las correspondientes a la señales Step (salto) y DIRC (dirección) del 1771. Como dijimos anteriormente, son las dos únicas señales que tiene el 1781 para controlar los motores paso a paso de control de la cabeza de lectura-escritura. Las patillas 17 y 18, correspondientes a PH3 y 3PM en el 1771 se denominan aquí OTSTR y AMOT.
— OTSTR (Out strobe, patilla 17). Esta señal tiene una doble misión según trabajemos en simple o doble

Bit	*Ordenes Tipo I*	*Leer identificador*	*Leer*	*Leer pista*	*Escribir*	*Escribir pista*
7	Disco no preparado	Disco no preparado	Disco no preparado	Disco no preparado	Disco no preparado	Disco no preparado
6	Protegido en escritura	0	Tipo de grabación	0	Protegido en escritura	Protegido en escritura
5	Cabeza enganchada	0	Tipo de grabación	0	Fallo en escritura	Fallo en escritura
4	Error en la búsqueda	MID no encontrada	Sector no encontrado	0	Sector no encontrado	0
3	Error CRC	Error CRC	Error CRC	0	Error CRC	0
2	Pista 0	Pérdida de dato	Pérdida de dato	Pérdida de dato	Pérdida de dato	Pérdida de dato
1	Indice	DRQ	DRQ	DRQ	DRQ	DRQ
0	Ocupado	Ocupado	Ocupado	Ocupado	Ocupado	Ocupado

Tabla 11.11 Significado de los bits del registro de estado

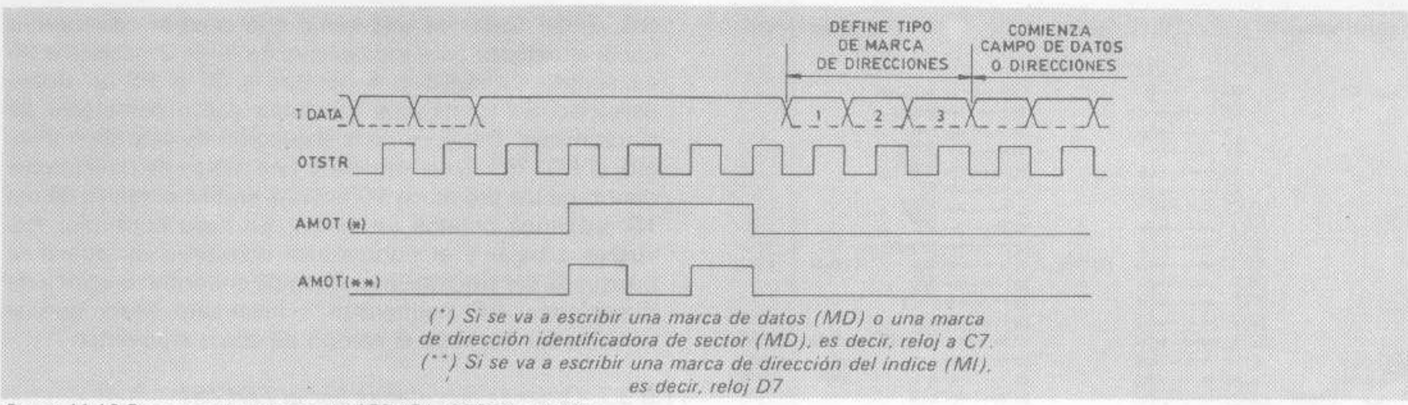


Figura 11.16 Diagrama de tiempos de OTDATA, OTSTR y AMOT

densidad. En simple densidad esta patilla saca la señal compuesta de datos y reloj para ser grabada en el disco (equivalente a la señal WD del 1771). En doble densidad esta señal es un reloj que indica, en cada flanco ascendente, el centro del bit de la señal que sale por OTDATA (Out data, patilla 31) para ser grabada en el disco.

— AMOT (Address Mark out, patilla 18). Esta señal es solamente usada cuando el controlador trabaja en doble densidad. Sirve para indicar a la circuitería externa que por la señal OTDATA (datos a grabar) va a transmitirse una marca de direcciones. Esta señal es necesaria para que la circuitería externa sepa qué marca de direcciones es y cuándo debe generarse para ser grabada en el disco, ya que el controlador sólo las genera cuando trabaja en simple densidad. En la figura 11.16 representamos los diagramas de tiempo de las señales OTDATA, OTSTR y AMOT.

— $\overline{\text{DDEN}}$ (Double density, patilla 37). Esta señal ocupa el lugar de $\overline{\text{DINT}}$ en el 1771. El nivel de esta señal programa al 1781 para trabajar en simple densidad ($\overline{\text{DDEN}}$ = 1) o doble densidad ($\overline{\text{DDEN}}$ = 0).

— OTDATA (Output data, patilla 31). Por esta patilla salen los datos para ser escritos en el disco cuando el controlador trabaja en doble densidad. El flanco ascendente de OTSTR indica el centro de cada bit. Los tres bits siguientes al flanco descendente de AMOT indican al circuito externo, generador de marcas de direcciones, el tipo de marca a generar según los códigos expresados en la tabla 11.12.

1	2	3	Marcas de direcciones
0	0	0	Datos borrados (MDB)
0	0	1	Marca de datos (MD)
0	1	0	Definida por el usuario
0	1	1	Marca de datos (MD)
1	0	0	Marca dirección del índice (MI)
1	0	1	No definida
1	1	0	Marca dirección del identificador del sector (MID)
1	1	1	No definida

Tabla 11.12 Código de las marcas

— INDATA (Input data, patilla 27). Correspondiente a FDDATA del 1771.
Por esta patilla, el circuito separador de datos externo presenta los datos al controlador 1781.

— INSTR (Input strobe, patilla 26). Corresponde a FDCLK del 1771. Los niveles altos de esta señal indican, aproximadamente, el centro de los datos en la señal INDATA.

— AMIN (Address mark detect in, patilla 25). Corresponde a la señal $\overline{\text{XTDS}}$ en el 1771. Esta señal es generada, al igual que las dos anteriores por el circuito separador de datos e indica al controlador que una marca de direcciones ha sido detectada. Los tres bits de datos que siguen al flanco descendente de AMIN contienen el código de la marca detectada tal como expresamos en la tabla 11.12. En la figura 11.17 representamos el diagrama de tiempos de las señales INDATA, INSTR y AMIN.

11.5.3 Controlador WD1791

Este nuevo controlador constituye una mejora de los dos anteriores fabricados por Western Digital (el WD1771 y el WD1781). Al igual que el 1781, el 1791 puede trabajar en simple y doble densidad pero además tiene incorporadas dentro de la misma pastilla funciones que no tenía el 1781 y que debían realizarse con circuitería externa. Entre dichas funciones destacamos el reconocimiento de los bits a ser grabados de forma que sacan al exterior señales para el circuito precompensador y generación y reconocimiento automático de las marcas de direcciones tanto en simple como en doble densidad.

11.5.3.1 *Señales del controlador*

Solamente describiremos el significado de las señales del

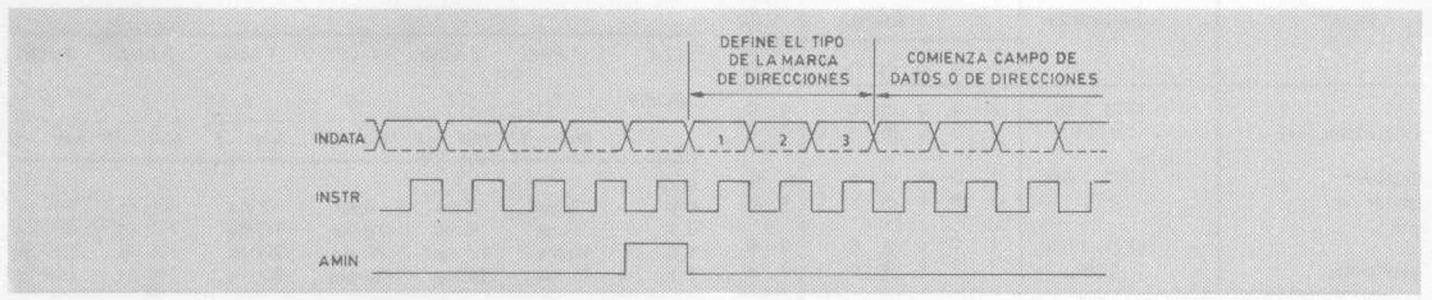


Figura 11.17 Diagrama de tiempos de INDATA, INSTR y AMIN

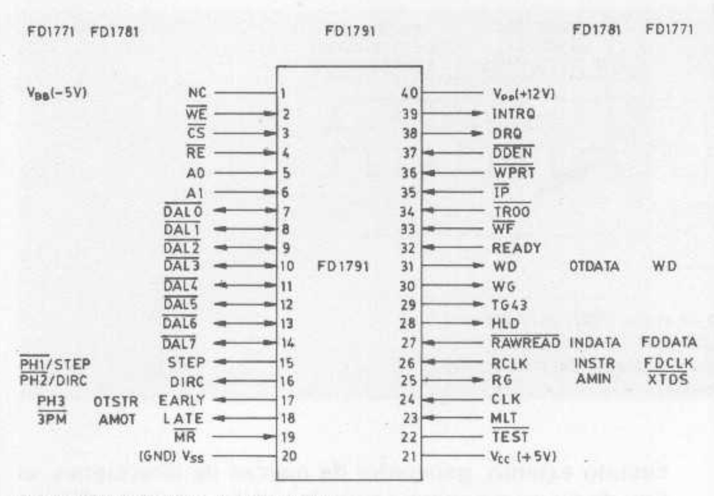


Figura 11.18 Señales del WD 1791

1791 que son diferentes a las correspondientes del 1771 o del 1781. En la figura 11.18 pueden verse las señales diferentes en cada uno de los controladores.

— EARLY y LATE (patillas 17 y 18).
Como ya hemos descrito en §11.2.3 es recomendable, al grabar en doble densidad, realizar una precompensación en aquellas secuencias de bits que originan desplazamientos de su posición normal en el momento de la lectura. Los desplazamientos que se han de dar (positivos y negativos) a los bits, dependen de la secuencia y del tipo de transporte de disco a usar. Por ello el controlador no realiza dentro esos desplazamientos y se limita a reconocer las secuencias de bits a grabar e informar a un circuito exterior si el bit que le está presentando en ese momento (por la patilla WD) debe o no desplazarse a la izquierda o a la derecha de su posición normal.
La señal EARLY indica que el bit que aparece por WD debe ser escrito antes. La señal LATE indica que debe escribirse más tarde. Si ambas señales están a nivel bajo indican que el bit actual debe grabarse en su posición normal. Volveremos a hablar de la precompensación al describir un circuito que la realiza.

— RAWREAD, RCLK y RG (patillas 27, 26 y 25 respectivamente). Son las señales necesarias para realizar las lecturas de datos.

— RAWREAD es la señal compuesta leída del disco (datos + reloj). Dicha señal debe tener un pulso negativo de 100 a 200 ns por cada pulso de datos o reloj.
RCLK (Read Clock) es una señal de reloj, generada por el circuito separador externo, que indica al controlador los cambios de flujo de la señal compuesta. La polaridad de esta señal no es importante pero sí los cambios de polaridad. Esta señal deberá ser cuadrada.

Orden	*Controlador*	*Código*							
		b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
Leer sector (es)	1771, 1781	1	0	0	m	b	E	0	0
	1791	1	0	0	m	x	E	0	0
Escribir sector (es)	1771, 1781	1	0	1	m	b	E	a_1	a_0
	1791	1	0	1	m	x	E	x	a_0
Leer pista	1771, 1781	1	1	1	0	0	1	0	S
	1791	1	1	1	0	0	1	0	x

Tabla 11.13 Códigos comparativos del WD1771 y WD1791

RG (Read Gate) es una señal que saca el controlador hacia el exterior para indicar que ha leído 2 octetos a 00 en simple densidad o 4 octetos a 00 o FF en doble densidad de forma que los toma como caracteres de sincronismo. Después de la detección de este sincronismo, el FD1791 debe encontrar una marca de direcciones dentro de los próximos 10 octetos en FM o dentro de los 16 próximos octetos en MFM. En caso contrario, RG vuelve a bajar y el controlador comienza de nuevo la búsqueda del sincronismo. Caso de encontrar la marca de direcciones, RG permanece a nivel alto hasta que el controlador lea todo el campo de datos siguientes.

11.5.3.2 *Ordenes ejecutadas por el FD1791*

En cuanto al número de órdenes diferentes que puede ejecutar, el 1791 tiene las mismas que el 1771 y el 1781. Sin embargo existen algunas pequeñas diferencias en cuanto al código y forma de ejecutar las órdenes relativas a leer y escribir datos en el disco.

En concreto, son diferentes los códigos de las órdenes para leer sector(es), escribir sector(es) y leer pista. En la tabla 11.13 ponemos los códigos comparados de estas tres órdenes para el 1771 (o 1781) y el 1791.

El bit 3 de las órdenes leer sector(es), escribir sector(es) tenía el significado de especificar formato IBM o formato libre. Sin embargo, el 1791 sólo trabaja con longitudes de sector compatibles con IBM, por lo que su valor no tiene ningún significado.

Los bits a_1, a_0 de la orden escribir sector(es) especificaban en el 1771 ó 1781 el tipo de marca de datos a grabar (puesto que se permitían 4 diferentes). Sin embargo, el 1791 sólo permite dos marcas de datos diferentes que se codifican en el bit a_0 ($a_0 = 0$ implica escribir la marca de datos FB y $a_0 = 1$ indica escribir la marca de datos borrados F8).

Por último el bit *S* de la orden leer pista no tiene ningún significado para el 1791 puesto que se vuelve a sincronizar automáticamente siempre que encuentra una marca de direcciones.

En cuanto a la ejecución de las órdenes, existen pequeñas diferencias en algunas de ellas con relación al 1771 o 1781. Por ejemplo, cada vez que el 1791 dé la orden de bajar la cabeza esperará 15 ms (frente a 10 ms en el 1771) antes de comprobar el estado de HLT; en cualquier orden de verificación estará intentando realizarla, si ésta falla, hasta encontrar 5 impulsos de índice (frente a 3 en el 1771); después de ejecutar cualquier orden, mantendrá la cabeza pegada al disco, si no recibe otra orden, durante 15 revoluciones (frente a 3 en el 1771). Las temporizaciones para el control del motor paso a paso están descritas en la tabla 11.14.

Durante la ejecución de la orden de escribir una pista (usada para formatear un disco virgen), el controlador genera diferentes señales compuestas al recibir un octeto de

CLX	*2 MHz*	*2 MHz*	*1 MHz*	*1 MHz*	*2 MHz*	*2 MHz*
DDEN	*0*	*1*	*0*	*1*	*x*	*x*
	test = 1	*test = 1*	*test = 1*	*test = 1*	*test = 0*	*test = 0*
r_1 r_0						
0 0	3 ms	3 ms	6 ms	6 ms	200 μs	400 μs
0 1	6 ms	6 ms	12 ms	12 ms	200 μs	400 μs
1 0	10 ms	10 ms	20 ms	20 ms	200 μs	400 μs
1 1	15 ms	15 ms	30 ms	30 ms	200 μs	400 μs

Tabla 11.14 Velocidad de salto entre pistas

Octeto transmitido al 1791	Interpretación del 1791 en FM ($\overline{DDEN}=1$)	Interpretación del 1791 en MFM ($\overline{DDEN}=0$)
00 - F4	Escribir 0 0 a F4 con reloj = FF	Escribir 00 a F4 en MFM
F5	No permitidio	Escribir A1(*) en MFM y reinicializar el CRC
F6	No permitido	Escribir C2(**) y reinicializar el CRC
F7	Generar los dos octetos del CRC	Generar los dos octetos del CRC
F8 - FB	Escribir F8-FB con reloj=C7 y reinicializar CRC	Escribir F8-FB en MFM
FC	Escribir FC con reloj=D7	Escribir FC en MFM
FD	Escribir FD con reloj=FF	Escribir FD en MFM
FE	Escribir FE con reloj=C7 y reinicializar CRC	Escribir FE en MFM
FF	Escribir FF con reloj=FF	Escribir FF en MFM

(*) Quitando el pulso del reloj entre los bits 4 y 5.
(**) Quitando el pulso de reloj entre los bits 3 y 4.

Tabla 11.15 Interpretación de la información a grabar

control según trabaje en simple o doble densidad. En la tabla 11.15 hemos representado dichas diferencias.

El contenido y significado del SR es el mismo que en el 1771 excepto el bit 6 de la orden leer que se pone a 0 en el 1791.

11.5.4 Controlador I8271

Es un controlador de la Intel que apareció en el mercado posteriormente que el WD1771. Está pensado para controlar discos sectorizados por software, en simple densidad y con formato compatible con el IBM 3740.

Sirviéndonos de comparación el WD1771, podemos enumerar las siguientes diferencias:
- Puede trabajar directamente con dos transportes de discos o con un transporte de doble cara.
- Está pensado para trabajar con DMA (acceso directo a memoria) de forma que además de las señales equivalentes a DRQ e INTRQ del 1771 posee otra entrada adicional ($\overline{DACK}$) que realiza automáticamente una transferencia controlada por el DMA entre el registro de datos del controlador y la memoria, o viceversa. Esta posibilidad simplifica el circuito de la interfaz.
- El conjunto de órdenes que puede ejecutar es superior al equivalente del 1771 en cuanto a número y potencia lógica de cada una de ellas.
- Sólo necesita una alimentación de +5 voltios.
- No tiene separador de datos internos por lo que la separación debe realizarse con circuitería externa mediante un monoestable o un PLO (oscilador enganchado en fase). El tipo de separación usada se le debe dar a conocer.
- Puede programarse el tiempo de salto entre pistas, el tiempo de bajada de la cabeza y el tiempo de estabilización de ésta. Por esta razón no necesita la señal equivalente a la HLT del 1771.
- Si no se trabaja por DMA, no existe ninguna señal externa que indique si debe realizarse una transferencia de datos. Para ello, se deberá leer el registro de estado y comprobar el valor de un bit.
- En cuanto al control del motor que controla la cabeza, puede realizarlo como el 1771 o suponiendo que el transporte ya tiene la lógica de secuenciamiento apropiado. En este caso, le indicará al transporte el número de saltos a realizar y el sentido de éstos.

A_1 A_0	$\overline{RE}=0$	$\overline{WE}=0$
0 0	R. de estado	R. de órdenes
0 1	R. de resultados	R. de parámetros
1 0	—	Reset
1 1	—	—

Tabla 11.16 Direccionado de los registros

Los registros que posee el controlador pueden clasificarse en dos tipos: aquellos directamente visibles al programador y otros que son usados por el microprograma que ejecuta las órdenes y que también pueden ser accedidos por el programador cuanto éste ordene ejecutar una orden especial equivalente a las que controlan el transporte del disco.

Los registros accesibles directamente al programador son: el registro de estado, registro de resultados, registro de órdenes, de parámetros y el registro de puesta en el estado inicial (Reset). Para direccionarlos se usan dos hilos de dirección y los correspondientes a lectura, $\overline{RE}$, y escritura, $\overline{WE}$. La tabla 11.16 indica la correspondencia entre ellos.

Las fases de una orden a ejecutar por el controlador son las siguientes: fase de la orden, fase de ejecución y fase de resultados. Durante la fase de la orden el programador debe escribir en el registro de órdenes el código correspondiente a la orden a ejecutar. Dependiendo del tipo de orden, puede necesitar desde 0 a 5 parámetros que deberán ser escritos en el registro de parámetros. La sincronización entre las escrituras de este registro debe realizarse a través del bit correspondiente en el registro de estado que indica si se puede o no escribir el siguiente parámetro. Una vez escrito el último parámetro necesitado, el controlador comienza a ejecutar la orden. Dicha ejecución puede necesitar la transferencia de datos. Como ya hemos dicho el controlador está preparado para realizarlas mediante DMA aunque también pueden realizarse bajo control de microprocesador. Al final de esta fase de ejecución el controlador envía una interrupción hacia el microprocesador. Durante la fase de resultados, el controlador informa al microprocesador del resultado de la operación a través del contenido del registro de resultados. Este registro contiene, codificados en cuatro bits, el resultado de la ejecución. La información es bastante explicativa en cuanto a los tipos de errores detectados.

Como ejemplos de órdenes vamos a describir las siguientes: inicialización, formatear una pista y exploración (Scan).

Inicialización. Esta orden es utilizada para definir las siguientes características básicas de funcionamiento: velocidad de salto entre pista y pista, tiempo de carga de la cabeza, tiempo de estabilización y tiempo que permanecerá la cabeza sobre la superficie del disco después de realizar la última orden.

El formato de dicha orden es el expresado en la figura 11.19 donde: CMD indica registro de órdenes; PAR, registro de parámetros; el valor del octeto de salto entre pistas indica los milisegundos requeridos para saltar de una pista a otra; el octeto de estabilización de la cabeza también indica el número de milisegundos requeridos (en el caso de los minidiscos estos tiempos se duplican); los 4 bits altos del

	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CMD	0	0	0	0	1	1	0	1	0	1
PAR	0	1	0	0	0	0	1	1	0	1
PAR	0	1	Salto entre pistas							
PAR	0	1	Estabilización de la cabeza							
PAR	0	1	Vueltas antes de levantar cabeza				Bajada de cabeza			

Figura 11.19 Formato de la orden de inicialización

último octeto escrito indican el número de vueltas que deberá estar la cabeza enganchada después de ejecutar la última orden (de 0 a 14 y si vale 15 indica que permanecerá siempre enganchada hasta recibir otra orden que la desenganchada), y los 4 bits bajos indican multiplicados por 4 el número de milisegundos que tarda la cabeza en bajar (estos tiempos se duplican en el caso de minidiscos).

Formatear una pista. La orden para formatear una pista simplifica bastante el control necesitado por el microprocesador si lo comparamos con el WD1771. De hecho no hay que suministrar caracteres especiales para que el controlador genere marcas de direcciones o CRC y, además permite programar la longitud de los gaps 1, 3 y 5 del formato IBM 3740.

El formato de la orden de formatear es el expresado en la figura 11.20 donde el nombre de cada campo indica el significado de su contenido excepto el campo longitud de sector, donde valores 0, 1,... 7 indican longitudes de sector de 128, 256, 512,... 16384 octetos.

Exploración. Este tipo de orden no está incorporada en los controladores de Western Digital o de Motorola, pero sí en el μPD765C de NEC. Al ejecutar esta orden el controlador compara una secuencia de datos (llave) que están en memoria con la secuencia que lee en el disco de forma que se acaba la ejecución cuando se cumple la condición de que la secuencia de datos de memoria sea mayor, menor o igual que la leída en el disco o cuando se termina de leer el último sector especificado para realizar la búsqueda. La comparación se hace con bloques del sector del mismo tamaño que la secuencia a comparar. Para realizar esta operación, debe emplearse un controlador de DMA (tal como el 8257 de Intel), trabajando en el modo de autocarga (auto-load mode).

Después de recibir la interrupción de final de orden, el microprocesador decodifica si se ejecutó o no correctamente. En caso afirmativo, el microprocesador puede mandar ejecutar una orden de lectura de los registros especiales donde encontrará información sobre el número del sector en

	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CMD	0	0	1	0	0	0	0	0	1	1
PAR	0	1	Número de la pista							
PAR	0	1	Longitud del gap 3 menos 6							
PAR	0	1	Longitud del sector				Sectores/pista			
PAR	0	1	Longitud del gap 5 menos 6							
PAR	0	1	Longitud del gap 1 menos 6							

Figura 11.20 Formato de la orden de formatear

	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CMD	0	0	1	0	0	0	0	S	0	0
PAR	0	1	Número de la pista							
PAR	0	1	Número del sector							
PAR	0	1	Longitud			Número de sectores a ser explorados				
PAR	0	1	Tipo de Exploración		Tamaño del salto					
PAR	0	1	Longitud, en octetos, de la llave							

Figura 11.21 Formato de la orden de exploración

D_7 D_6	*Comparación*
0 0	Llave = datos
0 1	Llave ⩾ datos
1 0	Llave < datos

Tabla 11.17 Tipo de exploración a realizar

el que se terminó la orden y el número de octetos que quedaban por leer de dicho sector en el momento del final de ejecución.

El formato de esta orden está expresado en la figura 11.21 donde el bit S indica la marca de los datos a ser comparados ($S = 0$ datos normales y $S = 1$ datos borrados), los dos bits del tipo de exploración codifican el tipo de comparación a realizar según indicamos en la tabla 11.17, los 6 bits del tamaño del salto especifican la distancia entre los sectores que van a ser comparados y el último octeto indica la longitud de la llave a comparar.

Además de estas tres órdenes descritas, el I8271 puede ejecutar otras órdenes del mismo tipo que el 1771, pero de mayor potencia.

11.5.5 Controlador MC6843

El MC6843 es un controlador de disco flexible fabricado por Motorola. A continuación daremos una serie de características generales de este controlador, comparándolas con las ya expuestas para el WD1771 e I8271.

— Está preparado para trabajar con formato IBM 3740, simple densidad y un transporte de una sola cara. En formato fijo, solamente admite 26 sectores de 128 octetos por pista.
— Las señales de interfaz para el microcomputador son muy específicas para trabajar con la familia de microprocesadores Motorola.
— Las transferencias de los datos pueden realizarse bajo control directo del microprocesador o con un controlador de DMA. Las señales de control del DMA están también preparadas para trabajar con el controlador fabricado por la misma casa, MC6844. Al igual que ocurre con el I8271 no se necesita generar la dirección física de los registros del controlador, al realizar la transferencia, lo cual minimiza el diseño.
— El número de registros internos es de 12, de forma que se necesitan tres hilos de direcciones y la señal R/W (lectura-escritura) para direccionarlos.
— El circuito sólo necesita una alimentación de + 5 voltios.
— No contiene separador de datos internos por lo que, al igual que el I8271, necesita obligatoriamente un circuito externo que realice esta función.
— Las señales para entrada de datos desde el transporte del

disco son análogas a las necesitadas por los controladores anteriores: una señal por donde entra la señal compuesta leída del disco, otra de reloj que es generada por el circuito separador de datos externo y una última señal de la salida del controlador que indica cuándo ha detectado sincronismo (2 octetos a 00 en simple densidad y 4 a 00 en doble densidad). Esta señal externa puede ser usada por el PLO separador de datos.
— Son programables, al igual que en el I8271, los tiempos de bajada de la cabeza, estabilización de la cabeza y número de vueltas que deben pasar después de ejecutar una orden antes de levantar la cabeza. Sin embargo, no se puede dejar la cabeza sobre el disco un tiempo indefinido. De hecho el manejo de la cabeza de lectura-escritura es menos flexible en este controlador frente a los otros dos.
— Las 10 órdenes que puede ejecutar tienen una potencia intermedia entre las ejecutadas por el WD1771 (menor potencia) y el I8271. El conjunto de órdenes es igual al del WD1771 y tal vez la variante más clara de reseñar sea la orden de formatear.
— La orden de formatear o escribir una pista admite en el 6843 dos variantes: formato fijo o variable. La primera de ellas significa que el microprocesador deberá dar al controlador los bits de datos y éste, intercalará pulsos de reloj FF. En la segunda de ellas el microprocesador debe dar al controlador la forma de la señal compuesta por lo que se puede escribir cualquier combinación de datos y reloj. Lógicamente la velocidad de intercambio de datos entre el microprocesador y el controlador será doble con el formato variable.

11.5.6 Controlador μPD765C

Es un controlador de NEC que puede trabajar con códigos de simple densidad (FM) o doble densidad (MFM) y formatos de pista compatibles con IBM.

Describiremos a continuación las características más importantes de esta pastilla, comparándolas con las ya expuestas al describir el WD1791.
— Solamente necesita una alimentación de +5 voltios.
— Está preparado para trabajar directamente con el controlador de DMA de Intel (8257) o bajo control del microprocesador. El controlador de DMA no necesita circuitería externa para generar las direcciones de los registros de datos del μPD765C y además éste tiene una señal de entrada correspondiente a la TC (cuenta terminada) del 8257. Esta señal finaliza con la orden que estaba ejecutando el PD765C de forma que permite controlar indirectamente el número de sectores a leer o escribir.
— Permite trabajar directamente con hasta 4 transportes de discos y con transportes de 2 caras.
— Algunas señales de control de transporte del disco están multiplexadas de manera que por una parte permite más señales de control con el mismo número de patillas pero, por la otra, necesita multiplexores externos para realizar la interfaz con el transporte.
— Como en el WD1791, no existe separador interno de datos de forma que esta función se ha de realizar externamente.
— Al igual que el WD1791 da información al circuito precompensador de datos para que éste sepa cómo desplazar los datos.
— Es programable, por programa, para trabajar con FM y MFM mientras que el WD1791 se programa mediante una señal externa, lo cual hace que la interfaz sea más cara.
— Contiene como el M6843 una señal para desbloquear los fallos de escritura del transporte.
— El programador solamente ve tres registros internos: el de estado, datos de lectura y datos de escritura, porque sólo necesita una línea de direcciones y otra de lectura-escritura para acceder a ellos.

La tabla 11.18 indica el direccionado de dichos registros.

A_0	Lectura	Escritura
0	Estado	
1	Dato a leer	Dato a escribir

Tabla 11.18 Direccionado de los registros

El registro de estado sirve para mandar órdenes al controlador, parámetros necesarios para esas órdenes y recibir resultados y estado del controlador después de que éste ejecute las órdenes.
— La secuencia de ejecución de las órdenes es equivalente a la descrita para el I8273. Así pues, se enviará la orden con los parámetros correspondientes al controlador, éste la ejecutará y presentará los resultados e información del estado hacia el microprocesador.

Esta estructura de funcionamiento es análoga a la empleada por el controlador de procedimientos de línea de Intel (8273) y la empleada por los controladores de NEC. De hecho el controlador de NEC para simple densidad, denominado μPD372 (no descrito en este capítulo) es análogo al I8271.
— Por lo general, las órdenes ejecutadas por el μPD765C cubren a las del WD1791 con la particularidad de que son más potentes (requieren menos control directo del microprocesador) y hay algunas que no están implementadas en el WD1791 (una de ellas es la orden de exploración, análoga a la descrita para el I8271).
— Característica propia de este controlador es que extiende el concepto de pista, para discos de una cara, al de cilindro, para discos de doble cara, de forma que permite la ejecución de órdenes de lectura-escritura sobre multisectores pertenecientes al mismo cilindro.
— El μPD765C puede realizar órdenes de buscar una pista con cuatro transportes de disco simultáneamente.
— Al igual que en el I8271 podemos definir, en una operación de escribir pista (usada generalmente para formatear el disco) el tamaño del gap 3 y el tipo de datos a escribir. El único control necesitado, por parte del microprocesador es de 4 octetos por sector.
— Permite trabajar directamente con sectores que contienen datos válidos o datos borrados.

11.5.7 Otros controladores integrados

Debido a la gran aceptación de los discos flexibles, existen actualmente muchos fabricantes que han lanzado al mercado controladores integrados. Pensamos que en adelante los constructores integrarán dentro de la misma pastilla algunas de las funciones que se tienen que realizar actualmente con circuitería externa, tal como la separación de datos, y otras nuevas tales como códigos de grabación programables, de forma que la fiabilidad de la interfaz completa sea más fácil de diseñar y más fiable.

Debido a la falta de espacio no hemos descrito otros controladores ya existentes y nos limitaremos a poner una

Compatib. bus proces.	Fabric.	Modelo	Veloc.	Formatos compatib.	Número de transportes	Característic.	Tecnología encapsul.	Comentario
8080	INTEL	2271	2 MHz 3 MHz	IBM 3740	2-4	Sectorización software multisector	NMOS+5 V 40 pin	Puede utilizarse con el controlador de DMA 8257
6800	Motorola	6843	1 MHz	IBM3740			NMOS 40 pin	Opción de DMA
6800	Motorola	6849	1 Mbyte/s	todos	1	FM,MFM, doble cara	NMOS+5 V 48 pin	DMA interno
9900	TI	9909	2 Mbyte/s	IBM	1-4	sectoriz. software o hardware, pre-compensación.	NMOS+5 V 40 pin	
PPS4 PPS8	Rockwell	10936	199 kHz 256 kHz	IBM	4		PMOS 42 pin (QIL)	
General	WD	1771	1-2 MHz	IBM3740	1-4		NMOS- 40 pin	
General	WD	1781	0,5-1-2 MHz	IBM 34	1	Doble densidad	NMOS 40 pin	
General	WD	1791/93	1-2 MHz	IBM 3740 IBM 34	1	Doble densidad Sectorización software	NMOS + 5 V 40 pin	
General	NEC	μPD 372	2 MHz	IBM 3740	16		NMOS+5 V 40 pin	
General	NEC	μPD765	8 MHz	IBM 34	4	Doble densidad multisector	NMOS+5 V 40 pin	Puede utilizarse con el controlador de DMA 8257
General	Signetics	8×330	1 MHz	IBM 3740 IBM 34	1-4		Bipolar I^2L	
General	SMC	7003	1 MHz	IBM 3740	4	Simple y doble densidad	NMOS +5V 40 pin	
General	SMC	3400	500 kHz		2	Sectoriz. hard-ware simple y do-ble densidad	NMOS+5 V 12 V 40 pin	
General	National	8291/3	1-2 MHz	IBM 3740 IBM 34	1	Sectoriz. software simple y doble densidad	XMOS+5 V 40 pin	Compatible con WD 1791/93

Tabla 11.19 Tabla comparativa de controladores integrados

tabla comparativa de algunos de ellos (tabla 11.19). La referencia de las características están obtenidas de la publicación EDN en su número del 20 de octubre de 1979.

11.6 EJEMPLO APLICACION 1

En este apartado vamos a describir un circuito que hace de interfaz entre un sistema microcomputador que tiene como bus el denominado MUBUS (capítulo 3) y el transporte de disco flexible SA800 de Shugart. Además del circuito expondremos los programas necesarios para leer y escribir un sector. El diseño ha sido realizado en el Laboratorio de Ordenadores de la E.T.S.I.T. de Barcelona.

11.6.1 Diseño del circuito

El esquema general de bloques del circuito de interfaz es el representado en la figura 11.22.

Como vemos es el típico de cualquier interfaz de periférico realizado usando un controlador integrado.

El bloque interfaz 1 tiene como misión adaptar las señales del bus del sistema a las señales necesitadas o generadas por el controlador integrado.

El bloque interfaz 2 realiza las mismas funciones entre el controlador y el periférico (en este caso el transporte del disco flexible).

A continuación describimos cada una de las interfaces 1 y 2 por separado.

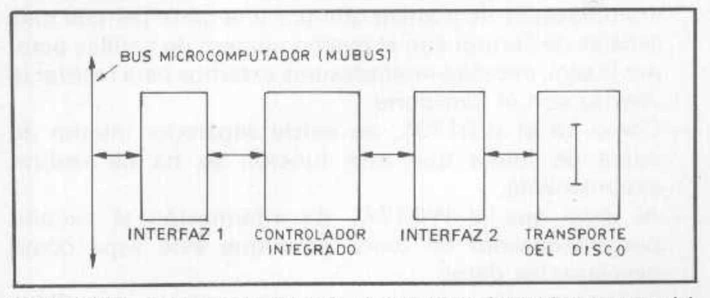


Figura 11.22. Interfaz entre bus del microcomputador y el transporte del disco

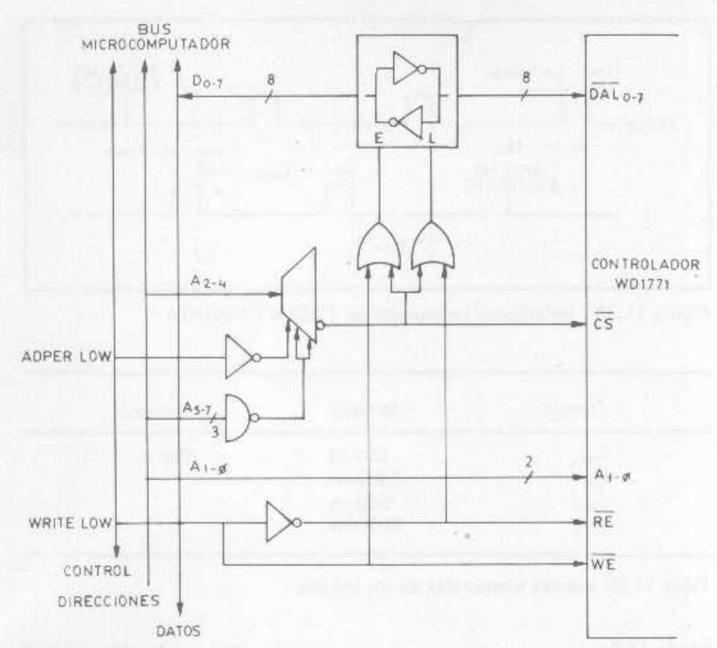


Figura 11.23 Comunicación de datos

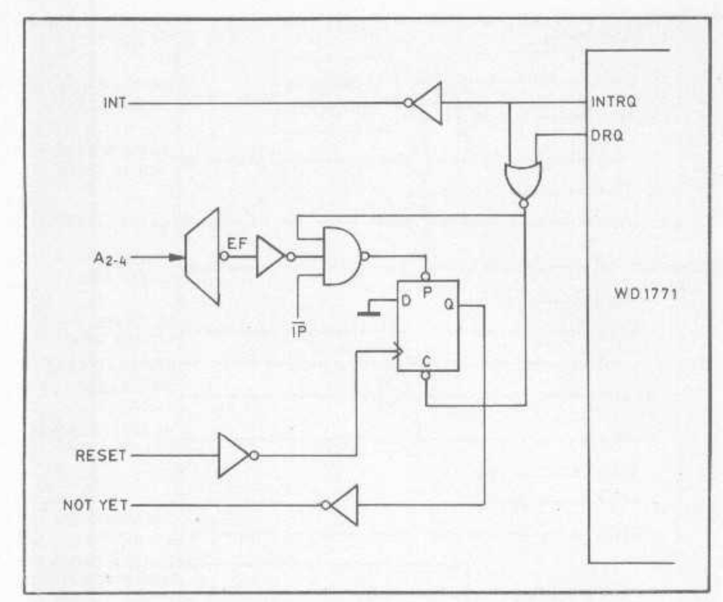


Figura 11.24 Circuito de sincronización

11.6.1.1 *Interfaz 1*

Podemos distinguir tres partes: la relativa a la comunicación de datos entre los registros del controlador y el microprocesador, la sincronización para el envío y recepción de esos datos y el final de ejecución de las órdenes y la necesaria para la reinicialización y reloj interno del controlador.

11.6.1.1.1 Comunicación de datos

El circuito montado para esa comunicación es el expresado en la figura 11.23.

Con relación a los datos hemos puestos unos buffers inversores tres estados bidireccionales ya que el controlador necesita los datos invertidos. Tal como hemos decodificado las direcciones los registros han quedado en las siguientes direcciones: E0 = CR, E1 = TR, E2 = SR y E3 = DR. Las señales $\overline{RE}$ y $\overline{WE}$ las obtenemos a partir de Write Low de MUBUS. El sentido de conducción del buffer de datos viene controlado por las señales $\overline{CS}$, $\overline{RE}$ y $\overline{WE}$.

11.6.1.1.2 Señales de sincronización

Recordemos que DRQ se pone a nivel activo siempre que se deba realizar una transferencia de un octeto entre el microcomputador y el controlador, o viceversa, y que INTRQ se pone a nivel activo siempre que el controlador termina la ejecución de una orden.

En la figura 11.24 representamos la forma de conectar ambas señales al bus del microprocesador. La señal INTRQ la unimos a la señal de interrupción de MUBUS. En cuanto a DRQ podrían existir varias soluciones. Una de ellas sería la de que el microprocesador estuviera comprobando su valor continuamente para saber cuando debe realizar las transferencias con el controlador. Respecto a esta solución debemos subrayar que para códigos de simple densidad (FM) se debe transferir un octeto cada 32 μs. La mayoría de los microprocesadores tienen tiempo suficiente para realizar esta operación. Sin embargo, en códigos de doble densidad la velocidad de transferencia es de 16 μs por octeto, velocidad que es demasiado grande para microprocesadores tipo 8080 o MC6800 de 1 MHz. Por esto en el diseño que hemos realizado para un controlador de doble densidad (ver apartado 11.6) hemos asociado la señal DRQ a un canal de acceso directo a memoria (DMA) de forma que éste realiza las transferencias entre el controlador y la memoria o viceversa.

La solución que hemos adoptado en este caso concreto es la siguiente: cada vez que el microprocesador realiza un envío o recepción de un octeto de datos, activa la señal EF del decodificador de forma que la salida *Q* del biestable se pone a 1 por lo que el microprocesador queda en un estado de espera (Wait) como si se tratara de un acceso a una memoria lenta. Las señales DRQ e INTRQ ponen, por separado, la salida *Q* a 0 sacando al microprocesador del estado de espera de forma que éste realiza otra transferencia con el controlador y se vuelve a poner en estado de espera. La señal INTRQ desbloquea de nuevo dicho mecanismo de espera y envía al microprocesador una interrupción.

Como nota adicional diremos que la señal $\overline{IP}$ también desbloquea el mecanismo de espera.

11.6.1.1.3 Reinicialización y reloj del controlador

Para realizar estas funciones se conecta la señal reset del MUBUS a la señal $\overline{MR}$ del controlador y el reloj se obtiene a través de la fase del microprocesador enviada por MUBUS.

11.6.1.2 *Interfaz 2*

En la figura 11.25 representamos el esquema de esta interfaz. La mayoría de las conexiones son directas entre señales del controlador y transporte del disco (salvo un cambio de nivel).

Para la sincronización de la carga de la cabeza hemos puesto un monoestable que se dispara cada vez que el controlador ordena bajarla. El tiempo del monoestable es ligeramente superior al tiempo necesitado para bajar la cabeza (35 ms en el SA800) de forma que pasado éste la señal HLT se pone a nivel alto indicando al controlador que ya puede comenzar las operaciones de lectura-escritura.

Tal como expresamos en la figura 11.25, la separación de los impulsos de datos y reloj ha sido realizada externamente al controlador. Normalmente los controladores pueden

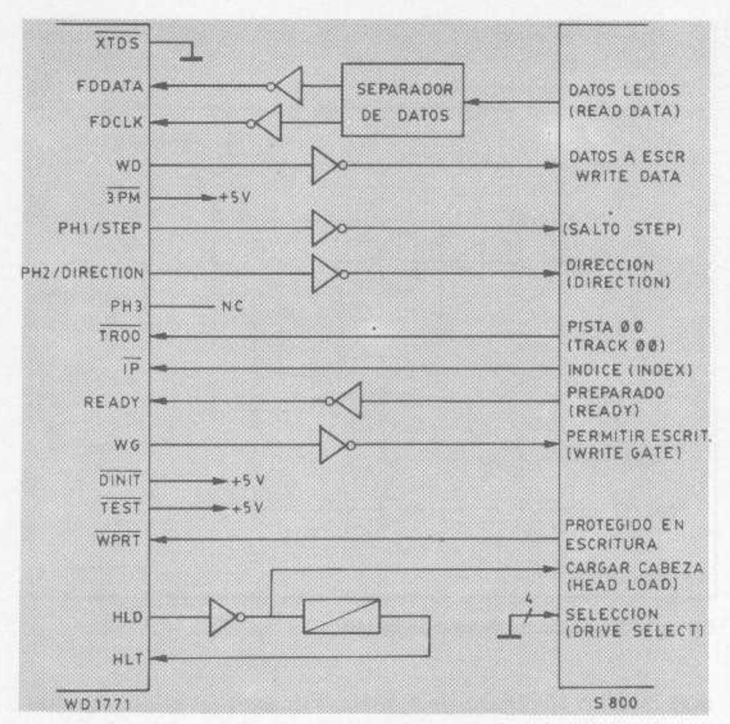


Figura 11.25 Interfaz Controlador-Transporte del disco

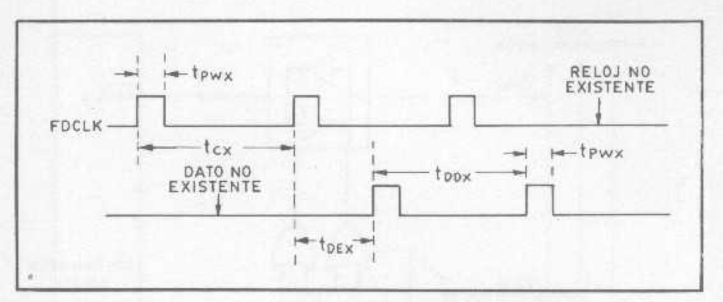


Figura 11.26 Relaciones temporales de FDCLK y FDDATA

Tiempo	*Mínimo*	*Máximo*
t_{PWX}	150 ns	350 ns
t_{CX}	2500 ns	
t_{DEX}	500 ns	
t_{DDX}	2500 ns	

Tabla 11.20 Valores temporales de las señales

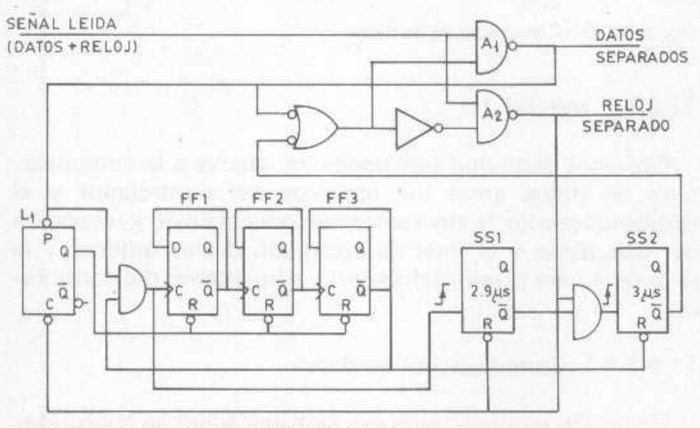


Figura 11.27 Circuito separador de datos y reloj

realizar también esta operación aunque, sin embargo, la mayoría de los fabricantes recomiendan que se realice externamente. Por otra parte, aunque los transportes de los discos también suelen poseer circuitos separadores, creemos que es recomendable hacerlo con un circuito adicional. Este circuito, debe tener como entrada la señal compuesta (datos-reloj) leída del disco y debe dar como salida las señales de datos y reloj independientes y de forma que las relaciones temporales entre ellas están dentro de los límites impuestos por el controlador integrado.

En la figura 11.26 expresamos los límites temporales, de las señales de reloj y datos, permitidos por el WD1771.

Los valores límites de los tiempos expresados en la figura están representados en la tabla 11.19.

El circuito empleado en nuestro diseño para la separación de los datos y reloj es el expresado en la figura 11.27.

En la figura 11.28 vemos los diagramas de tiempo de las señales. Suponemos que al principio llegan al circuito un conjunto de impulsos de reloj sin impulsos de datos (correspondientes a los últimos octetos de sincronismo dentro de los gap. Ver formato IBM 3740) y que después viene el octeto correspondiente a la marca dirección de los datos (MD). Este octeto constituye la secuencia más crítica para sincronizar puesto que las células de los bits 2, 3 y 4 no contienen impulsos de reloj. Sin embargo vemos que el circuito no pierde el sincronismo.

Algunas consideraciones que nos ayudan a explicarnos el funcionamiento del circuito son las siguientes:

— En ausencia de impulsos de datos, los impulsos de reloj

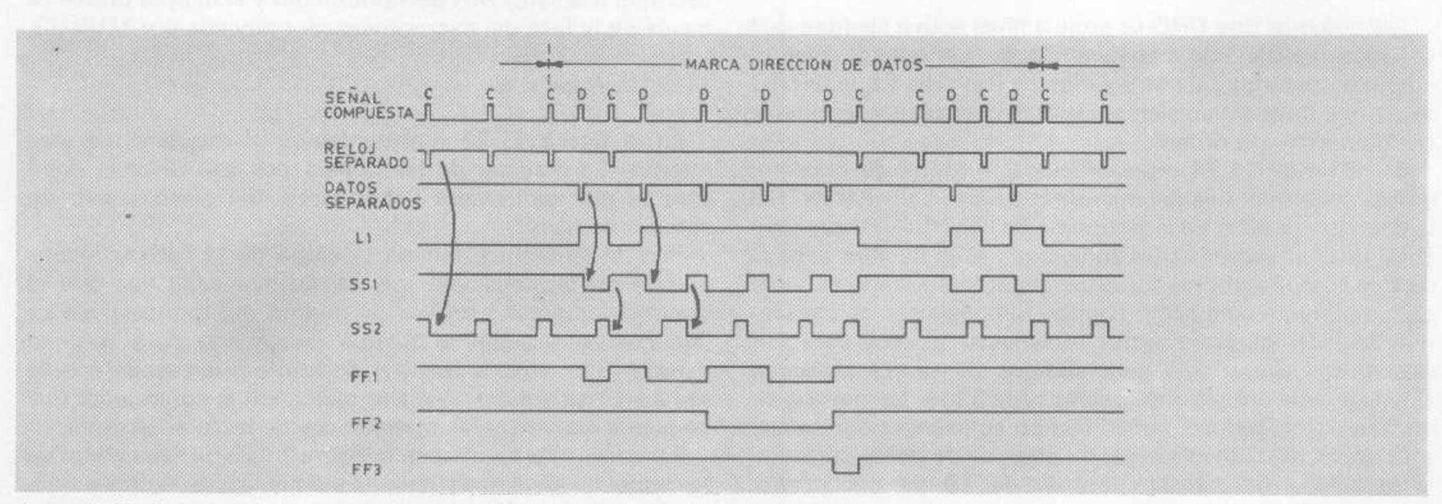


Figura 11.28 Diagrama de tiempos de las señales del circuito separador de datos y reloj

salen por la puerta NAND 2, invertidos en relación a como entran en el circuito.

— El flanco ascendente de cada uno de los impulsos de reloj de la salida, dispara el monoestable SS2 (de aproximadamente 3 μs de duración) cuya función es abrir una ventana en la puerta NAND 1 de forma que si durante este tiempo llega algún impulso de dato pueda pasar, con polaridad invertida, a la salida de datos separados.

— El nivel bajo de los impulsos de datos a la salida pone a 1 el biestable L1 y el flanco ascendente activa por una parte el biestable FF1 y el monoestable SS1 (de aproximadamente 2,9 μs de duración).

— El biestable L1 sirve para controlar la cadena de biestables FF1, FF2 y FF3. Como vemos L1 es puesto a uno a cada impulso negativo de datos en la salida y es puesto a cero a cada impulso negativo del reloj, de salida. Si la salida L1 está a 0, también lo están FF1, FF2 y FF3.

— En ausencia de impulsos de reloj en la señal compuesta, los impulsos de datos disparan el SS1 de forma que el flanco ascendente de SS1 vuelve a disparar el SS2. Este mecanismo permite abrir ventanas para los datos en ausencia de los impulsos de reloj.

— El contador formado por FF1, FF2, FF3 actúa en ausencia de impulsos de reloj y sirven para detectar la ausencia de más de tres impulsos de reloj consecutivos.

Por último representamos en la figura 11.29 el circuito completo salvo la parte separadora de datos y reloj, donde A_1 y A_3 son 74240, A_2 es un 74138, A_4 es un 7474, A_5 un 74123 y E1 un 81LS95. El interruptor I, situado junto a la pastilla E1 sirve para programar manualmente el código de Restart de servicio de interrupción. (En el diseño del programa de formatear hemos supuesto un Restart 7.) El biestable A_4 de la derecha sirve para seleccionar dos transportes de disco.

11.6.1.3 *Ejemplos de programas*

En general la ejecución de cualquier orden de las once que puede cumplimentar el controlador tiene las siguientes fases: inicialización, ejecución y comprobación.

Durante la fase de inicialización se preparan los registros necesarios de la UCP (ejemplo: puntero al inicio de la tabla de memoria donde se colocarán los datos leídos de un sector) y se envían a los registros del controlador las informaciones necesarias (ej.: el SR se cargará con el número del sector que se quiera leer). Esta fase termina con el envío del código de la orden a ejecutar al registro de órdenes (CR).

Una vez el controlador ha recibido el código de la orden a ejecutar mandará las señales necesarias al transporte del

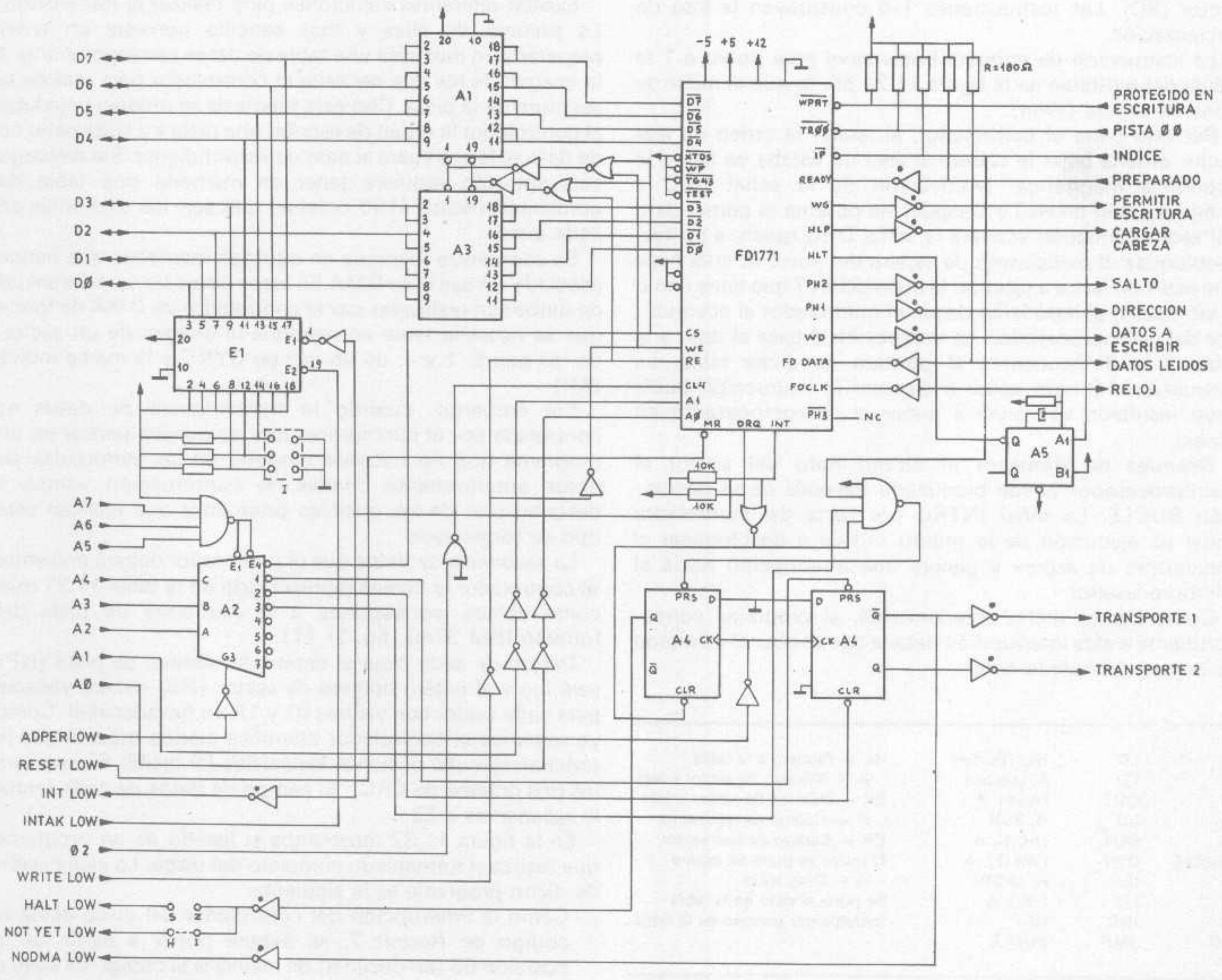


Figura 11.29 Esquema general del circuito

disco para realizar la operación. Algunas órdenes (ej.: lectura o escritura de un sector) necesitan, tal como hemos visto, una sincronización entre el microprocesador y el controlador para las transferencias de los datos. Otras tales como la de búsqueda de pista no lo necesitan. La fase de ejecución termina con el envío, por parte del controlador, de la señal INTRQ hacia el microprocesador.

En la última fase de la orden se debe ejecutar un programa que compruebe el resultado de la ejecución de la orden. Para ello se deberá comprobar el contenido del registro de estado (SR) del controlador.

Como ejemplos de aplicación vamos a describir las fases de inicialización y ejecución de las órdenes lectura y escritura de un sector.

11.6.1.4 *Lectura de un sector*

En la figura 11.30 está el listado de uno de los posibles programas que realizan esta función. Suponemos que la cabeza de lectura-escritura está sobre la pista deseada.

La instrucción 1 sirve para cargar en los registros H,L el puntero de la tabla de memoria donde se van a meter los datos leídos del disco. Las instrucciones 2 y 3 cargan en el registro SR del controlador el número del sector que se quiere leer. Las instrucciones 4 y 5 envían al registro de órdenes del controlador (CR) el código de la orden leer un sector (8C). Las instrucciones 1-5 constituyen la fase de inicialización.

La instrucción de etiqueta bucle sirve para poner a 1 la salida del biestable de la figura 11.24 por lo que el micro se pone en espera (Wait).

Por otra parte el controlador, al recibir la orden de leer sector ordena bajar la cabeza si ésta no estaba ya sobre la superficie magnética. (Activación de la señal HLD y comprobación de HLT.) Después de obtener el primer dato del sector requerido activará la señal DRQ quien, a su vez, desbloquea el mecanismo de espera del micro lo cual hace que éste comience a ejecutar la instrucción 7 que tiene como misión pasar el dato leído desde el controlador al acumulador del microprocesador. La instrucción 8 pasa el dato a la tabla y la 9 incrementa el puntero de dicha tabla. La instrucción 10 hace saltar a ejecutar la instrucción bucle cuyo resultado es volver a poner al microprocesador en espera.

Después de transferir el último dato del sector el microprocesador queda bloqueado después de la instrucción BUCLE. La señal INTRQ por parte del controlador (final de ejecución de la orden) vuelve a desbloquear el mecanismo de espera y genera una interrupción hacia el microprocesador.

Como hemos dicho anteriormente, el programa correspondiente a esta interrupción deberá comprobar el resultado de la ejecución de la orden.

```
1       LD     HL, (Buffer)   ; HL ← Puntero a la tabla
2       LD     A, (sector)    ;  A ← Número de sector a leer
3       OUT    (ASR), A       ; SR ← Número de sector a leer
4       LD     A, 8CH         ;  A ← Código de leer sector
5       OUT    (ACR), A       ; CR ← Código de leer sector
BUCLE   OUT    (WAIT), A      ; El micro se pone en espera
7       IN     A, (ADR)       ;  A ← Dato leído
8       LD     (HL), A        ; Se pone el dato en la tabla
9       INC    HL             ; Incrementar puntero de la tabla
10      JMP    BUCLE
```

Figura 11.30 Programa de lectura de un sector

```
1       LD     HL, (Buffer)
2       LD     A, (sector)
3       OUT    (ASR), A
4       LD     A, OAH;        A←Código de escribir sector
5       OUT    (ASR), A
BUCLE   LD     A, (HL)
7       OUT    (WAIT), A      ; Micro en espera
8       OUT    (ADR), A       ; DR←dato a escribir
9       INC    HL
10      JMP    BUCLE
```

Figura 11.31 Programa de escritura de un sector

11.6.1.5 *Escritura de un sector*

El programa de escritura de un sector es muy parecido al de lectura y su listado está expuesto en la figura 11.31.

11.6.1.6 *Formateado del disco*

Se entiende por formatear un disco virgen al proceso de grabar sobre cada una de las pistas toda la información excepto los datos de los sectores. Es decir, si por ejemplo queremos formatear un disco en formato IBM 3740, deberemos escribir en cada pista y en secuencia correcta los campos correspondientes a dicho formato (fig. 11.11).

Existen diferentes soluciones para realizar el formateado. La primera de ellas y más sencilla consiste en tener preparada en memoria una tabla de datos correspondiente a la imagen de los que necesita el controlador para realizar la escritura de la pista. Con esta hipótesis se ordenará ejecutar al controlador la orden de escribir una pista y a cada petición de dato se le entregará el dato correspondiente. Sin embargo esta solución requiere tener en memoria una tabla de aproximadamente 4156 octetos, que son los existentes en cada pista.

Se economiza memoria en aquellas interfaces que tienen asociado un canal de DMA En estos casos las transferencias de datos son realizadas por el controlador de DMA de forma que se necesita tener en memoria la imagen de un sector, de los gap 5, 1 y 4, de un campo SYNC y la marca índice (MI).

Sin embargo, cuando la transferencia de datos es controlada por el microprocesador es posible pensar en un programa que no necesite ninguna de las estructuras de datos anteriormente citadas. A continuación vamos a describir uno de los posibles programas que realizan este tipo de formateado.

La secuencia de datos que el procesador deberá presentar al controlador la hemos representado en la tabla 11.21 que, como vemos, corresponde a la estructura de pista del formato IBM 3740 (fig. 11.11).

Dentro de cada pista el octeto de número de pista (NP) será fijo y el octeto número de sector (NS) deberá variarse para cada sector con valores 01 y 1A en hexadecimal. Como ya sabemos el controlador identifica ciertos octetos que le ordenan ejecutar acciones especiales (al recibir F7 escribirá los dos octetos de CRC). El campo de datos de cada sector lo rellenamos a E5.

En la figura 11.32 mostramos el listado de un programa que realiza el formateado completo del disco. La explicación de dicho programa es la siguiente:

— Como la interrupción del controlador del disco envía el código de Restart 7, se deberá poner a partir de la posición 56 (en decimal) de memoria el código de salto a la dirección donde comienza la subrutina que detecta si la

```
Programa principal

1       MVI     A,0C3H
2       STA     RST7
3       LXI     H, VERR
4       SHLD    RST7+1

5       MVI     A 0D0H
6       OUT     CR
7       CALL    PISTA 0
8       XRA     A
9       STA     PISTA

FPISTA: MVI     A,01H
11      STA     SECTOR
12      CALL    FORM
13      CALL    STEPIN
14      JC      FPISTA
15      RST     0

        Pista 0

PISTA 0: MVI    B,90H
17      MVI     A,0AH
18      EI
19      OUT     CR
20      HLT
21      RET

Salto a la pista contigua interior

STEPIN: LXI     H, PISTA
23      INX     H
24      MOV     A,M     ; última pista grabada
25      CPI     4DH     ; última pista
26      RNC
27      MVI     B,0
28      MVI     A,5AH
29      EI
30      OUT     CR      ; Step-in
31      HLT
32      RET

Subrutina de error

VERR    IN      SR      ; Leer STR
34      ANA     B       ; Máscara con B
35      STC
36      RZ              ; Orden correcta
ERROR:  MOV     B,A
38      PUSH    B
39      MVI     C,#
40      CALL    CO      ; Sacar por TTY #
41      POP     B
42      POP     H
43      RST     0

Formateado de una pista

FORM    MVI     B,64H
45      MVI     A,0F4H
46      EI
47      OUT     CR
48      XRA     A
49      MVI     C,2EH   ; Gap 5+SYNC
FORM1:  OUT     WAIT
51      OUT     DR
52      DCR     C
53      JNZ     FORM1
54      MVI     A,0FCH  ;MI
55      OUT     WAIT
56      OUT     DR
57      XRA     A
58      MVI     C,20H   ; Gap 1+YNC
FORM2:  OUT     WAIT
60      OUT     DR
61      DCR     C
62      JNZ     FORM2
FORM3:  MVI     A,0FEH
64      OUT     WAIT
65      OUT     DR      ; MID
                        (continúa)⟶
66      LDA     PISTA
67      OUT     WAIT
68      OUT     DR
69      XRA     A
70      OUT     WAIT
71      OUT     DR      ; NC
72      LDA     SECTOR
73      OUT     WAIT
74      OUT     DR      ; NS
75      INR     A
76      STA     SECTOR
77      XRA     A
78      OUT     WAIT
79      OUT     DR      ; L
80      MVI     A, 0F7H
81      OUT     WAIT
82      OUT     DR      ; CRR
83      MVI     C,11H
84      XRA     A
FORM4:  OUT     WAIT
86      OUT     DR      ; Gap 2+SYNC
87      DCR     C
88      JNZ     FORM4
89      MVI     A,0FBH
90      OUT     WAIT
91      OUT     DR      ; MD
92      MVI     C,80H
93      MVI     A,E5H
FORM5:  OUT     WAIT
95      OUT     DR      ; Datos a E5
96      DCR     C
97      JNZ     FORM5
98      MVI     A,0F7H
99      OUT     WAIT
100     OUT     DR      ; CRC
101     XRA     A
102     MVI     C,21H
FORM6:  OUT     WAIT
104     OUT     DR      ; Gap 3+SYNC
105     DCR     C
106     JNZ     FORM6
107     LDA     SECTOR
108     CPI     1BH
109     JNZ     FORM3   ; Más sectores
110     XRA     A
FORM7:  OUT     WAIT
112     OUT     DR      ; Gap 4
113     JNZ     FORM7
114     RET
```

Figura 11.32 Programa de formatear

orden de escribir pista se realizó o no correctamente. Dicha subrutina comienza en la dirección VERR. (Instrucción 33 del programa.) Las instrucciones 1-4 realizan esta operación.

- Las instrucciones 5 y 6 sirven para enviar al controlador la orden de interrupción forzada de forma que se pone inactiva la señal de interrupción que estaba a nivel activo después del Reset general del sistema (ver órdenes ejecutadas por el WD1771).
- La instrucción 7 llama a la subrutina de posicionar la cabeza sobre la pista 00 (PISTA 0). Esta subrutina envía el código de la orden RESTORE (instrucción 19) después de poner en B la máscara 90 H que será empleada por la subrutina VERR para detectar si la orden RESTORE se ejecutó o no correctamente.
- Si la orden de ir a la pista 00 se realizó correctamente, se ejecutarán las instrucciones 8 y 9 que tienen como misión poner en la dirección de memoria PISTA el valor 0 (Pista que se va a formatear a continuación.) Las instrucciones 10 y 11 ponen en la dirección SECTOR el valor 1 correspondiente al primer sector.
- La instrucción 12 sirve para llamar al programa FORM que se encarga de formatear una pista. Dicho programa

Significado	Número de octetos	Valor	Comentario
GAP5	40	00 ó FF	
SYNC	6	00	
MI	1	FC	; Escribir marca índice
GAP1	26	00 ó FF	
SYNC	* 6	00	
MID	1	FE	; Escribir marca identificadora
NP	1	00 a 4C	; Número de pista
NC	1	00	
NS	1	01 a 1A	; Número de sector
L	1	00	
CRC	1	F7	; Escribir CRC
GAP2	11	00 a FF	
SYNC	6	00	
MD	1	FB	; Escribir marca de datos
DATOS	128	E5	; Datos a E5
CRC	1	F7	; Escribir CRC
GAP3	27	00 a FF	
GAP4	** 247	00 a FF	

* Esta secuencia se repite 26 veces
** Se continúa escribiendo hasta detectar el pulso del índice.

Tabla 11.21 Secuencia de datos para formatear en IBM 3740

- contiene las instrucciones 44-114 de forma que es fácil identificar cada uno de los campos de la tabla 11.21.
— Si se realizó bien el formateado de una pista, se vuelve a la instrucción 11 que a su vez, llama a la subrutina STEPIN (salto a la pista contigua interior). Dicha subrutina devuelve el valor del bit de acarreo a 0 si ya se formateó la última pista en cuyo caso la instrucción 15 vuelve el control al monitor (RST 0) y el formateado ha sido acabado. En caso contrario se vuelve a llamar al programa FORM.
— La subrutina VERR (instrucciones 33-43) comprueba si la última orden ejecutada se realizó o no correctamente. En caso negativo se envía por el teletipo el carácter '#' indicativo de que existe algún error. La subrutina CO, no definida en el listado saca por el teletipo el carácter contenido en el registro C.
— Como último punto diremos que este programa graba todos los gaps a 00.

11.7 EJEMPLO APLICACION 2

En lo que sigue vamos a describir, brevemente, la realización de un posible circuito controlador de un disco flexible trabajando en doble densidad, MFM, con formato de pista IBM 34.

En concreto, describiremos dos posibles realizaciones para la interfaz 1 (la primera igual que la equivalente en el ejemplo aplicación 1 y la otra con un controlador de DMA), un circuito empleado para la precompensación de los datos a grabar y un circuito separador de datos y reloj. Las partes descritas pertenecen a un diseño realizado en la E.T.S.I. de Telecomunicación de Barcelona en el Laboratorio de Ordenadores.

11.7.1 Interfaz 1

Como parámetro importante para realizar esta interfaz debemos recordar que la velocidad de transferencia de datos entre el sistema microcomputador y el controlador del disco es de 500.000 bits/s equivalente a 1 octeto cada 16 μs.

Si empleamos un microprocesador que pueda realizar los bucles de lectura-escritura de un octeto en este tiempo, (M6800 de 2 MHz) podemos emplear el mismo diseño ya descrito en el ejemplo aplicación 1. Sin embargo, si el microprocesador no es lo suficientemente rápido, deberán realizarse las transferencias mediante un canal de acceso directo a memoria.

Si empleamos un circuito controlador de DMA (tipo 8257 de Intel), la señal DRQ del WD1791 se unirá a la DRQ de un canal del 8257 de forma que cada vez que se necesite una transferencia de un octeto entre el controlador del disco y la memoria, o al revés, será el controlador de DMA quien la realice. El I8257 necesitará además un 8212 para generar las direcciones altas de memoria y puesto que el WD1791 no está directamente preparado para trabajar por DMA necesitaremos lógica adicional para generar las señales correspondientes de selección del registro de datos del controlador del disco. Los bloques anteriores son los necesitados para controlar cualquier periférico con un canal de DMA usando el I8257. El lector interesado en la forma de conectar estos bloques deberá consultar el manual de periféricos de Intel.

Además de la lógica asociada al canal de DMA se deberá realizar, al igual que en el ejemplo 1, la circuitería necesaria para que el microprocesador pueda direccionar los registros de ambos controladores (DMA y disco).

Puesto que el WD1791 debe ser programado externamente para trabajar en simple o doble densidad (señal DDEN), se deberá emplear un biestable para mantener dicha señal al valor requerido. El valor de la salida del biestable será cambiado por el microprocesador mediante programa.

11.7.1.2 *Circuito precompensador de datos*

Al hablar de la escritura de la señal compuesta en doble densidad, hemos visto la necesidad de que ciertos bits necesiten una precompensación para facilitar su correcta recuperación al leerlos.

El diseño del circuito que realiza esta precompensación depende de las señales que saca el controlador indicando qué tipo de precompensación se debe de realizar (adelantar o retrasar el bit) y de las características del transporte del disco que indican qué valores son razonables para realizar esos desplazamientos.

El algoritmo de decidir que bits deben retrasarse o adelantarse en la escritura es realizado internamente por el WD1791 de acuerdo con la tabla 11.22 y es indicado al circuito precompensador mediante las señales EARLY (avance) y LATE (retardo).

Bits anteriores	*Bit presente*	*Bit próximo*	*Avance/retardo*
x 1	1	0	avance
x 0	1	1	retardo
0 0	0	1	avance
1 0	0	0	retardo

Tabla 11.22 Decisión para la precompensación

Early	*Late*	*Retardo*
1	0	0
0	0	Δ
0	1	2Δ

Tabla 11.23 Desplazamientos en el precompensador

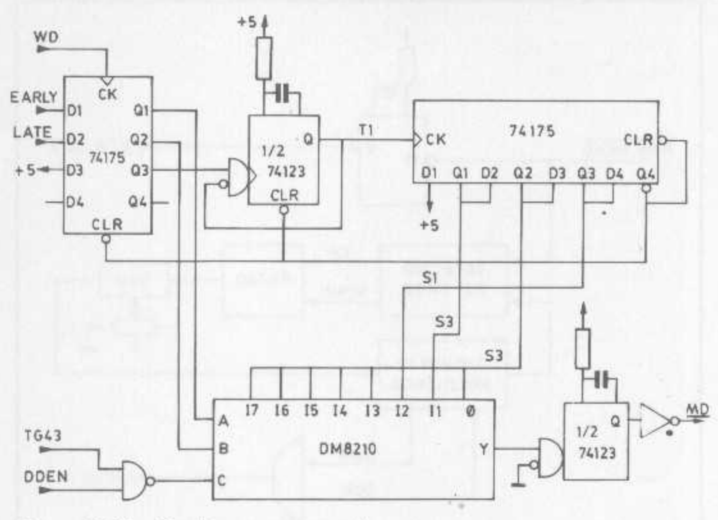


Figura 11.33 Circuito precompensador

Para que el circuito precompensador pueda adelantar los impulsos requeridos deberá retrasar sistemáticamente todos aquellos bits que no requieran precompensación en el valor Δ, no retardar los que necesitan ser adelantados y retrasar 2Δ aquellos que necesitan ser retrasados. Es decir, la tabla de funcionamiento del circuito precompensador será la expresada en la tabla 11.23.

Un posible circuito que realiza esta precompensación es el representado en la figura 11.33. El monoestable M1 fija el valor de precompensación (en este caso a 200 ns) y el monoestable M2 fija la anchura del pulso a grabar (en este caso 250 ns). Ambos valores son especificados por el fabricante del transporte y están en función, sobre todo, de las características de la cabeza de lectura-escritura.

Las entradas a dicho circuito son los datos a grabar (WD), las señales EARLY y LATE y los estados de las señales TG43 (pista mayor que la 43) y DDEN (grabación en doble densidad). Ambas señales deberán valer 1 para que se realice la precompensación. Como salida está la señal $\overline{WD}$ que se une directamente a la señal WRITE DATA (datos a escribir) del transporte SA800.

El funcionamiento del circuito es el siguiente:

— A cada flanco de subida de la señal WD se cargan los estados de las señales Early y Late en dos biestables del 74175, y en Q_3 el valor 1. El flanco de subida en Q_3 dispara el monoestable M1 cuya duración es la del tiempo de precompensación, Δ. La conexión realizada en este monoestable hace que el flanco de bajada de su salida lo vuelva a disparar, es decir, funciona como un oscilador de período aproximado al tiempo del pulso que genera.

— Los flancos de subida de la salida de este monoestable actúan sobre el segundo 74175 que está conectado para realizar un contador. Cuando el 4 bit de este contador, Q_4, se pone a 1, la señal Q_4 inhibe por una parte a M1 y pone a cero las salidas del primer 74175. Debido a la conexión realizada en el segundo 74175, sólo se permitirá generar tres impulsos al M1 que indicarán los posibles retardos a introducir en la señal. Dichos retardos están también representados en las señales S1, S2 y S3 (salidas del segundo 74175).

S1, S2 y S3 son tres de las ocho entradas del multiplexor de forma que la salida en cada momento de este multiplexor depende de los valores de Early y Late y $\overline{PCEN}$ (que indica si se debe o no realizar precompensación).

— El flanco de subida en la salida Y del multiplexor dispara el monoestable M2 que fija la anchura de los pulsos a grabar.

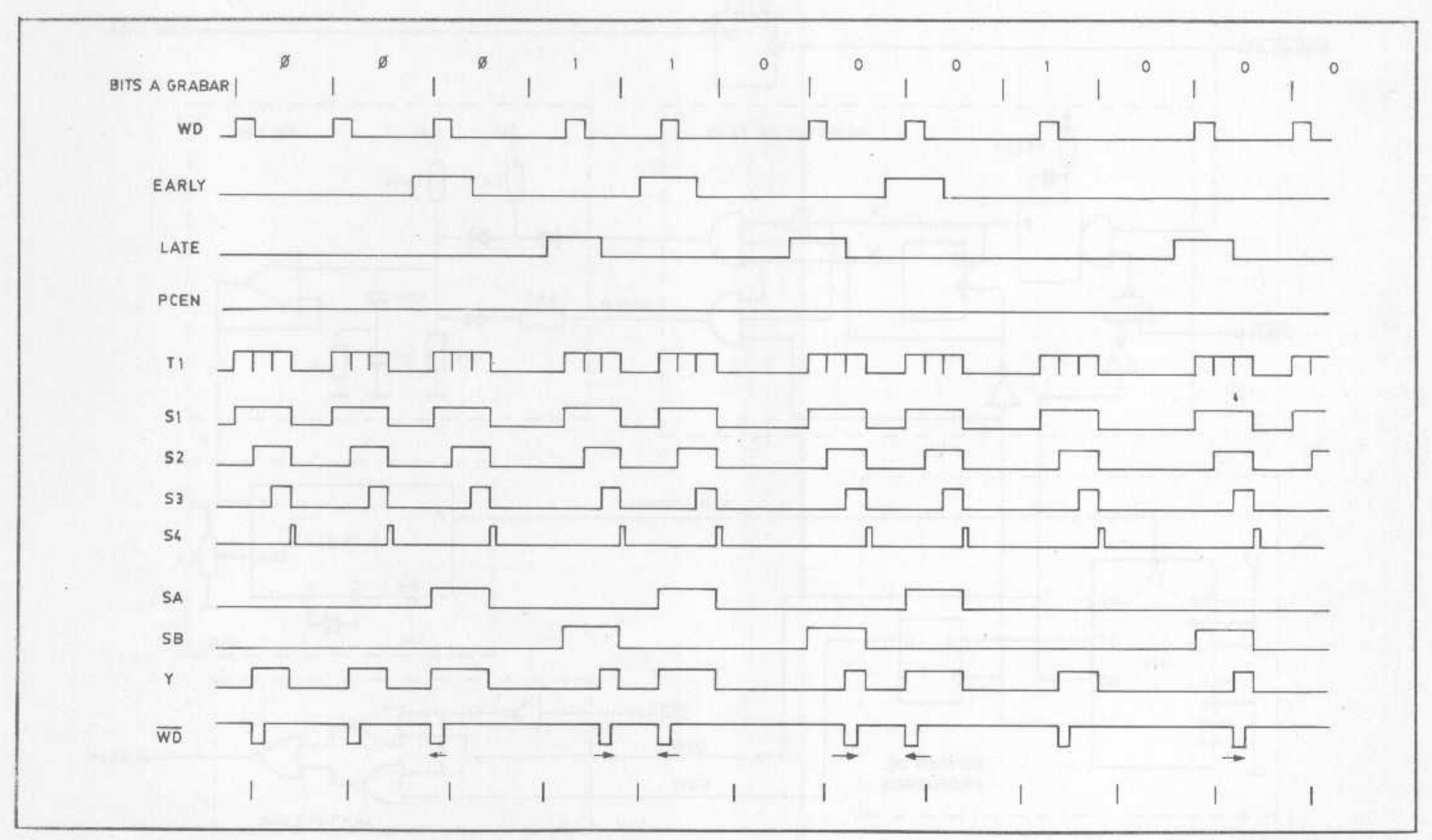


Figura 11.34 Diagramas de tiempo de las señales del circuito precompensador

	Mínimo	Típico
t_{PN}	100 ns	200 ns
t_{bC}	1600 ns	2000 ns
tx_1	40 ns	
tx_2	40 ns	
T_a	800 ns	1 μs
T_b	800 ns	1 μs
T_c	1600 ns	2 μs

Tabla 11.24 Valores típicos de las señales

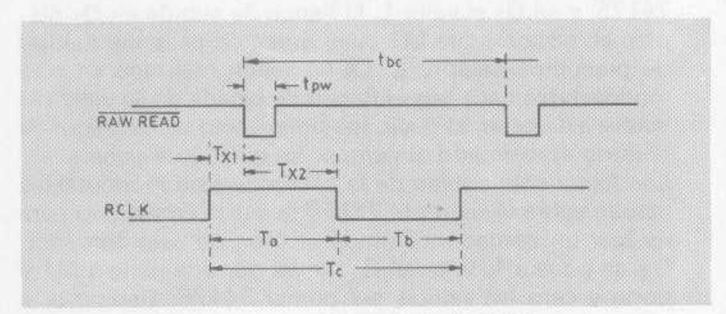


Figura 11.35 Señales RCLX y RAW READ del WD1791

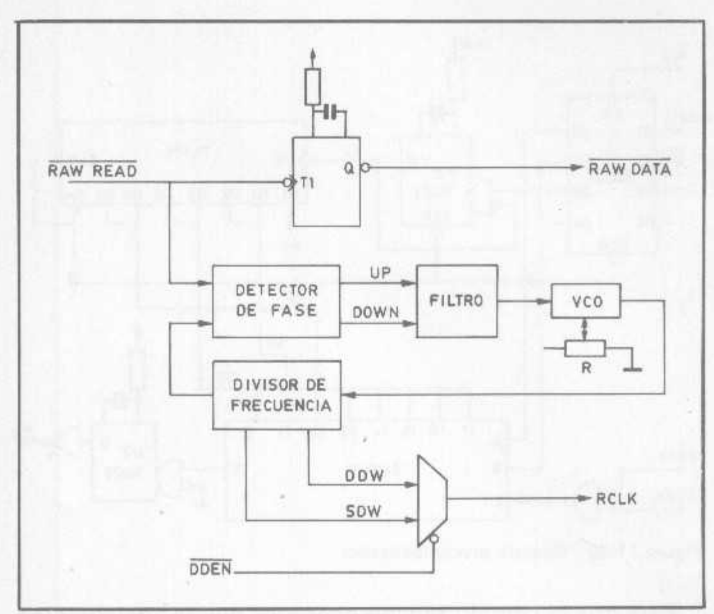


Figura 11.36 Diagrama de bloques del reparador de datos

En la figura 11.34 se representa el diagrama de tiempos de algunas de las señales del circuito.

11.7.1.3 *Circuito separador de datos*

Tal como hemos visto al describir los controladores integrados de doble densidad WD1791 y μPD765C, la separación de los datos debe realizarse externamente. El diseño del circuito dependerá de las señales necesitadas por el controlador y es aconsejable realizarlo basándonos en un PLO (oscilador enganchado en fase).

El diagrama de tiempos de las señales RCLK y RAW READ necesitadas por el WD1791 es el expresado en la figura 11.35, y los valores de los tiempos en la tabla 11.24.

La señal RCKL debe ser una onda cuadrada y estar en fase con los pulsos de la señal RAW DATA. El desfasaje entre

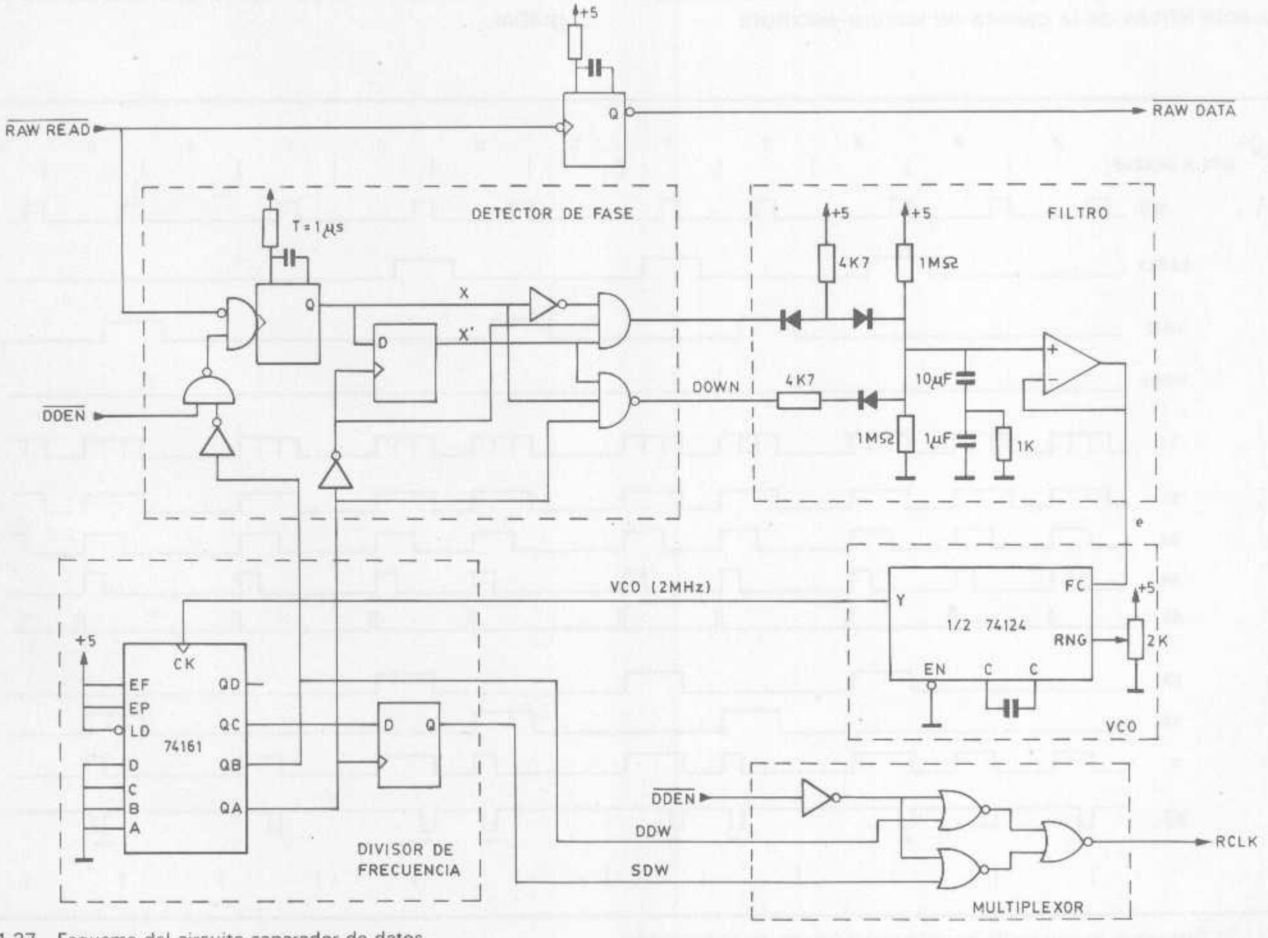


Figura 11.37 Esquema del circuito separador de datos

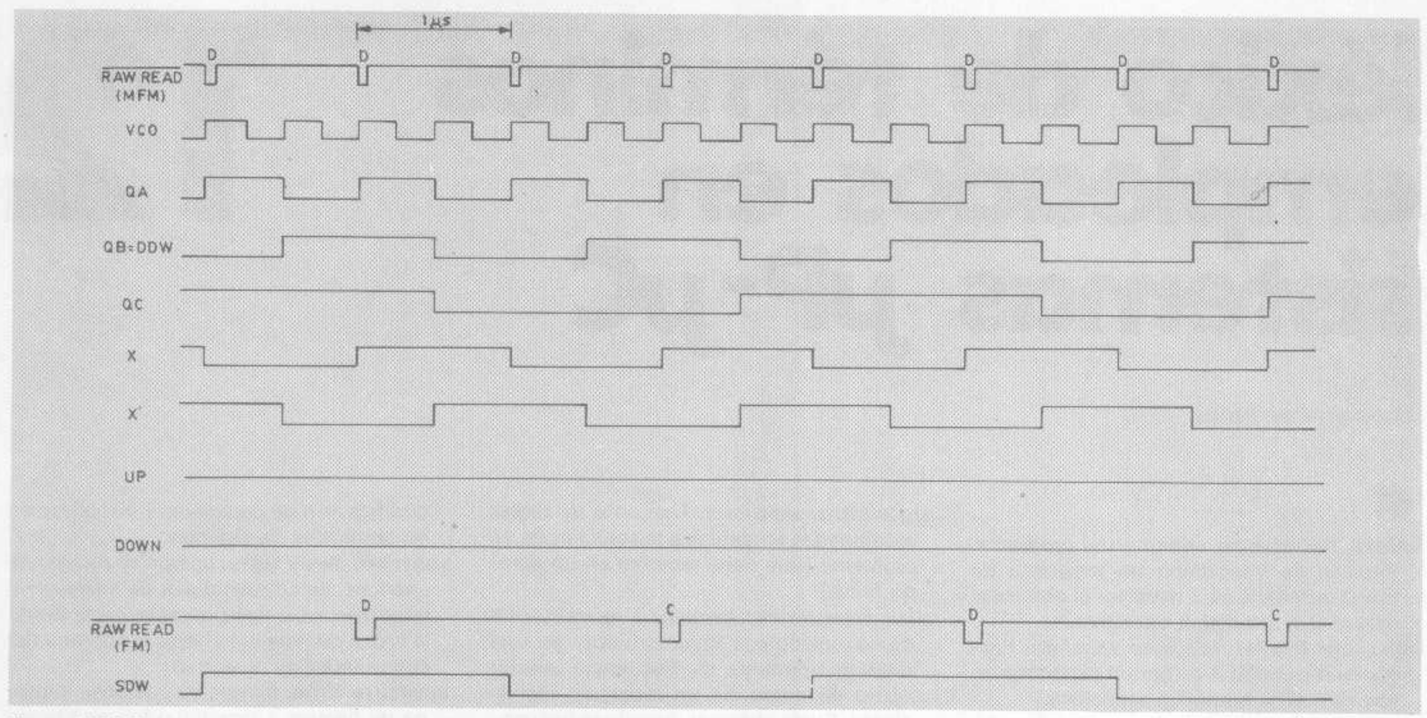


Figura 11.38 Señales del sincronizador de lectura cuando el oscilador está en fase con la señal de entrada

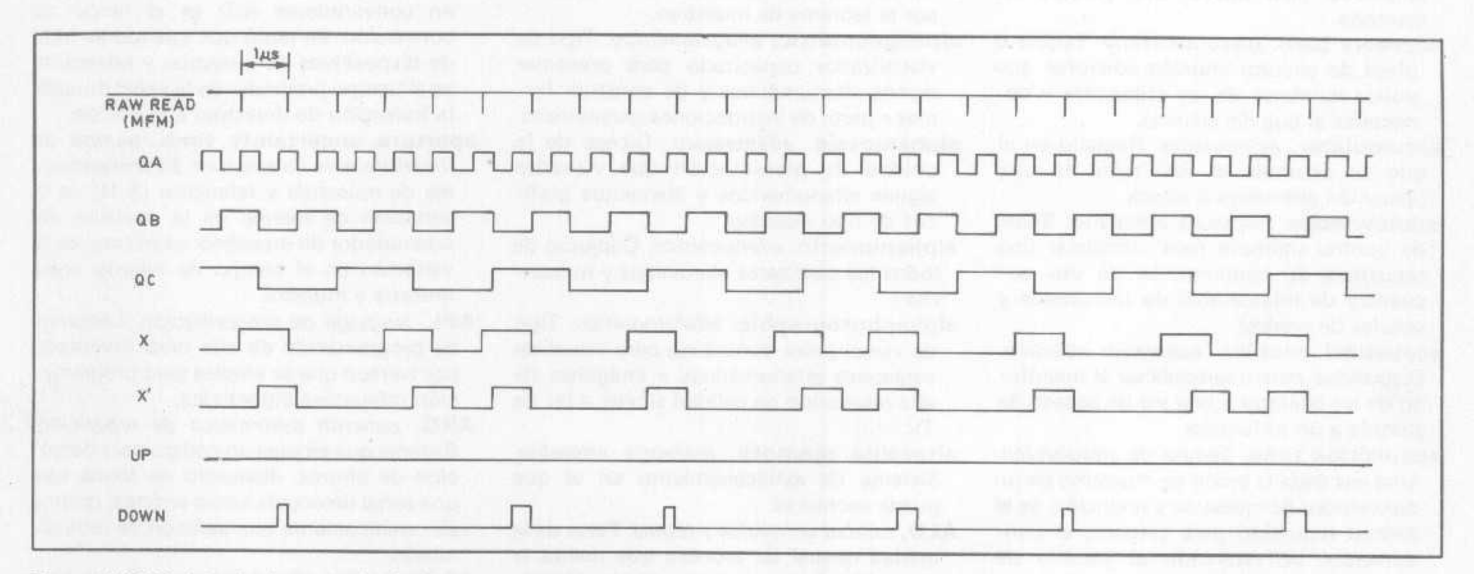


Figura 11.39. Señales de sincronizador de lectura con el oscilador a frecuencia más elevada que la nominal

estas dos señales es bastante crítico cuando se trabaja en doble densidad.

El diagrama de bloques utilizado en el separador está expresado en la figura 11.36.

Como vemos consta de un oscilador controlado por tensión (VCO) cuya frecuencia central es ajustable a partir de R. La señal de salida del oscilador, una vez dividida, es comparada en fase con la señal de entrada RAW READ de forma que el detector de fase proporciona como salida dos señales de impulsos UP y DOWN de duración proporcional al error de fase observado. Mediante un filtro se obtiene una señal de error *e*, cuya polaridad es función de las señales UP y DOWN y cuyo valor es proporcional a las duraciones de estas señales. La señal *e*, es realimentada al oscilador de forma que éste tiende a modificar su frecuencia para ponerse en fase con la señal de entrada. Del divisor de frecuencia se obtienen las señales DDW y SDW correspondientes a la señal RCLK para doble y simple densidad respectivamente que son seleccionadas por medio del multiplexor controlado por DDEN (doble densidad). El monoestable sirve para ajustar la anchura de los pulsos que entran al WD1791.

En la figura 11.37 se representa el esquema del circuito que realiza el separador de datos donde hemos recuadrado los bloques descritos en la figura 11.36. En las figuras 11.38 y 11.39 hemos puesto los diagramas de tiempo cuando el oscilador está en fase con la señal de entrada, figura 11.38 y cuando el oscilador funciona a frecuencia más elevada de la nominal, figura 11.39.

léxico de términos empleados en sistemas µP-µC

12

Eugenio Rey Veiga

a

abort. *terminación anormal del programa.* Proceso de interrumpir un programa de forma ordenada para devolver el control al operador o al sistema operativo.

absolute loader. *cargador absoluto.* Programa destinado a cargar un programa en una dirección numérica específica.

access time. *tiempo de acceso.* Tiempo empleado para buscar una palabra en memoria.

accesory card. *placa accesoria.* Tarjeta o placa de circuito impreso adicional que puede montarse en un ordenador y conectarse al bus del sistema.

accumulator. *acumulador.* Registro en el que se acumula el resultado de una operación aritmética o lógica.

acknowledge. *respuesta afirmativa.* Señal de control utilizada para completar una secuencia de confirmación de una secuencia de intercambio de indicativos y señales de control.

acoustical coupler. *acoplador acústico.* Dispositivo para interconectar el micrófono de un teléfono a una vía de acceso de entrada a un ordenador.

acquisition time. *tiempo de adquisición.* Una vez dada la orden de muestreo en un convertidor de muestreo y retención, es el tiempo requerido para cargarse el condensador de retención al cambio de tensión a fondo de escala y posteriormente permanecer dentro de una banda de error especificado en torno al valor final.

A/D. *analógico-digital.* Conversión de las tensiones y corrientes analógicas procedentes de un sensor para su representación digital en sistemas informáticos.

adder. *sumador.* Dispositivo cuya salida representa la suma de las cantidades que constituyen sus entradas.

add-on. *ampliación.* Circuitería o sistema que puede adjuntarse a un ordenador para ampliar memoria o prestaciones.

address. *dirección.* Número que indica la posición de una palabra en memoria.

address mark. *marca de dirección.* Código especial de 8 bits empleado en el comienzo de una zona específica de la pista de un disco.

ALGOL. *lenguaje de programación algorítmico.* Lenguaje de alto nivel que dispone de una estructura conocida como «exenta de contexto».

algorithm. *algoritmo.* Conjunto de reglas escalonadas tendentes a la solución de un problema (que debe terminar en un tiempo finito).

alias frequency. *frecuencia aparente.* En datos analógicos reconstruidos, es una falsa componente de frecuencia inferior como resultado de un muestreo insuficiente. Suele aparecer cuando la frecuencia de muestreo es inferior a la requerida por el teorema de muestreo.

alphageometric. *alfageométrico.* Tipo de visualizador capacitado para presentar signos alfanuméricos y de construir formas a partir de instrucciones geométricas.

alphamosaic. *alfamosaico.* Dícese de la unidad de presentación que visualiza signos alfanuméricos y elementos gráficos de tipo mosaico.

alphanumeric. *alfanumérico.* Conjunto de todos los caracteres alfabéticos y numéricos.

alphaphotographic. *alfafotográfico.* Tipo de visualizador concebido para visualizar caracteres alfanuméricos e imágenes de alta resolución de calidad similar a las de TV.

alterable memory. *memoria alterable.* Sistema de almacenamiento en el que puede escribirse.

ALU. *unidad aritmética y lógica.* Parte de la unidad central de proceso que realiza la ejecución de operaciones tanto lógicas como aritméticas.

analog/analogue *analógico.* Perteneciente a datos representados en forma de cantidades físicas continuamente variables que, si bien no se pueden contar, sí se pueden medir. Este concepto contrasta con el de digital. Por ejemplo, tensiones, corrientes, medidas angulares, etc.

analog multiplexer. *multiplexor analógico.* Disposición de conmutadores con una conexión de salida común para seleccionar una de entre varias entradas analógicas. La señal de salida sigue a la entrada seleccionada con un pequeño error.

analyser/analyzer. *analizador.* Cualquier dispositivo que controla un componente, tarjeta o sistema y representa los datos analizados.

AND. *Y.* Operación lógica definida por la regla siguiente: si A y B son 1, C es 1; de lo contrario, C es 0.

Antiope. *Antiope.* Normativa francesa para codificación de caracteres y visualización en terminales de videotex.

aperture delay time. *tiempo de retardo de apertura.* En convertidores de muestreo y retención, es el tiempo transcurrido desde la orden de retención hasta la apertura del conmutador de muestreo.

aperture time. *tiempo de apertura.* Ventana de tiempo o incertidumbre de tiempo en realizar una medida. Cuando se aplica en convertidores A/D es el timpo de conversión, en tanto que cuando se trata de dispositivos de muestreo y retención, es el tiempo promedio de la señal durante la transición de muestreo a retención.

aperture uncertainty time. *tiempo de incertidumbre de apertura.* En convertidores de muestreo y retención (S/H) es la variación de tiempo en la apertura del conmutador de muestreo; asimismo, es la variación en el tiempo de retardo entre muestra y muestra.

APL. *lenguaje de programación.* Lenguaje de programación de alto nivel inventado por Iverson que se emplea para programación interactiva algorítmica.

ARQ. *petición automática de repetición.* Sistema que emplea un código para detección de errores, dispuesto de forma que una señal detectada como errónea, origina automáticamente una petición de retransmisión.

ASCII. *código americano normalizado para el intercambio de información.* Código de caracteres utilizado por la mayoría de equipos distintos a los fabricados por IBM y Western Digital.

ASR. *transmisión-recepción automática.* Terminal que dispone de un perforador de cinta, lector de cinta, teclado e impresora.

assembler. *ensamblador.* Programa que toma la forma mnemónica del lenguaje simbólico y lo convierte en código objeto binario para su ejecución.

asynchronous. *asíncrono.* Evento o dispositivo que no es síncrono con la temporización de la unidad central de proceso (o de otros procesos).

attenuation. *atenuación.* Pérdida de amplitud de una señal.

auto-answer. *respuesta automática.* Dispositivo que confiere a las unidades de transmisión de datos, la facultad de responder de forma automática a una llamada recibida.

automatic calling unit. *dispositivo de llamada automática.* Dispositivo de marcación mediante el cual, un equipo de transmisión de datos puede generar automáticamente una llamada.
automatic storage allocation. *asignación de memoria automática.* Denominación que recibe la acción de asignar memoria a las variables automáticas.
auxiliary storage. *almacenamiento auxiliar.* Unidad de almacenamiento de datos distinta a la memoria principal.

b

background program. *programa subordinado.* En multiprogramación, dícese del programa de baja prioridad que opera cuando el procesador no trabaja con programas preferentes.
backplane. *conexionado posterior (entre tarjetas).* Area física donde se enchufan las tarjetas de un sistema. Normalmente contiene las barras del sistema tanto en forma de circuito impreso o cableado.
backup copy. *copia de seguridad.* Copia que contiene una reproducción o duplicado del conjunto de datos que se están utilizando, y que se guarda como reserva para casos de pérdida o destrucción del original.
bandwidth. *ancho de banda.* Frecuencia a la cual la ganancia de un amplificador o cualquier otro circuito se reduce en 3 dB de su valor de corriente continua. Asimismo es la gama de frecuencias en las cuales la atenuación es inferior a 3 dB a partir del valor de la frecuencia central.
base register. *registro de base.* Registro que contiene direcciones de base para referencias de tipo indexado. La dirección final se obtiene por adición de un desplazamiento.
BASIC. *código de instrucciones simbólicas de carácter general para principiantes.* Lenguaje de alto nivel inventado por Dartmouth con fines instructivos. Es fácil de aprender y de usar y es el que guarda mayor similitud con el FORTRAN.
batch. *lote.* Colección de documentos, registros o transacciones agrupadas con la finalidad de procesarlos como una sola unidad. Durante este proceso, no es posible realizar comunicaciones interactivas entre el programa y el usuario.
baud. *baudio.* Bits por segundo. Actualmente, unidad binaria de transmisión de información por segundo.
Baudot. *Baudot.* Código antiguo utilizado en teletipos de nivel 5 y en equipos de telex.
BCD. *decimal codificado en binario.* Representación de los números decimales en la que cada dígito se representa mediante 4 bits.
benchmark program. *programa patrón.* Programa específico para medir la velocidad de un ordenador en una posición bien definida, tal como una transferencia en serie, de multiplicar 8 por 8 bits.
bibliographic database. *base de datos bibliográficos.* Base de datos que contiene atributos de información acerca de una materia o fuente de información.
bibliographic record. *registro bibliográfico.* Colección de campos de datos tratados como una entidad lógica que describe una materia específica o fuente de información.
bi-directional. *bidireccional.* Flujo de datos que puede circular por un cable en cualquier sentido.
binary cell. *célula binaria.* Celdilla de almacenamiento capaz de retener un dígito binario.
binary number. *número binario.* Cantidad representada en un código binario; es decir, mediante un conjunto de unos y ceros.
binary search. *búsqueda binaria.* Técnica en la que el intervalo de búsqueda se divide por dos en cada iteración.
binary synchronous communication. *comunicación binaria síncrona.* Tipo de transmisión en el que se hace uso de la transmisión síncrona en binario.
binary synchronous transmission. *transmisión síncrona en binario.* Forma de transmisión en la que el sincronismo de los caracteres se controla por señales de tiempo.
binary to decimal conversion. *conversión binario a decimal.* Proceso de conversión de un número binario a su equivalente decimal.
bipolar. *bipolar.* Tecnología utilizada en la fabricación de circuitos integrados. Emplea elementos de conmutación de transistor y está basada en la utilización de portadores mayoritarios para conmutar y amplificar.
bistable. *biestable.* Dispositivo que siempre se encuentra en uno de los dos estados posibles.
bit. *bit.* Contracción de binary digit (dígito binario). Unidad de información que puede adoptar dos valores o estados distintos.
bit map. *mapeado por bits.* Método de creación de imágenes en la pantalla de un ordenador por asignación de un emplazamiento de memoria individual a cada punto en pantalla.
bit-slice microprocessor. *microprocesador expandible.* Componente que efectúa una separación de n bits en una CPU tradicional en la que normalmente $n = 4$. Un microprocesador expandible contiene todos los elementos de la CPU, incluyendo los multiplexadores, desplazadores, ALU, registros y acumuladores.
bits per inch. *bits por pulgada.* Medida que define la densidad de registro en una pista. Se abrevia bpi.
bit rate. *velocidad de transmisión.* Velocidad expresada en bits por segundo. Se abrevia bps.
blanking. *borrado.* Función de suprimir en una unidad de presentación visual, el haz de electrones a fin de evitar luminiscencia o brillo.
block. *bloque.* Unidad física de información en un registro lógico. Normalmente se expresa en bytes.
board. *tarjeta.* Placa de baquelita o de fibra de vidrio utilizada para montar los circuitos integrados. Las interconexiones pueden ser arrolladas, soldadas o impresas.
boolean algebra. *álgebra de Boole.* Algebra capaz de poner en forma de ecuaciones las proposiciones lógicas, cuyos factores de operación son AND (Y), OR(O), NOT (NO), NOR (NO-O), etc. Desarrollada por Georges Boole.
bootstrap. *programa inicializador.* Programa utilizado para poner en funcionamiento un ordenador. Usualmente pone a cero la memoria, pone a punto los dispositivos de E/S y carga el sistema operativo procedente de la ROM, disco o cassette.
branch. *bifurcación.* Instrucción idéntica a "jump" (salto); origina una transferencia de control a otra secuencia de programa.
breakpoint. *punto de ruptura.* Dispositivo de hardware o software que detiene un programa y vuelca el estado normal de la máquina.
bubble memory. *memoria de burbujas.* Dispositivo de almacenamiento que utiliza dominios magnéticos de tipo microscópico en un substrato de granate de aluminio.
buffer. *memoria intermedia, memoria tampón.* Memoria, sección de memoria o registro que recoge la información procedente de una memoria auxiliar, o más generalmente, de un dispositivo de E/S, reteniéndola para transferirla oportunamente a la memoria interna o principal del ordenador.
En un sentido amplio, dícese de cualquier memoria destinada a retener información durante un tiempo determinado, hasta el momento de la transferencia adecuada.
bug. *error.* Término general que indica la presencia de un error en un programa o el funcionamiento defectuoso del equipo.
built-in check. *verificación incorporada.* Dícese de determinados dispositivos o técnicas especiales de control y comprobación automática que van incluidos en diversos equipos.
bulk storage. *almacenamiento de gran capacidad.* Sistema de almacenamiento masivo, cuyo tiempo de acceso es superior al de la memoria principal a la que complementa.
bump. *memoria anexa.* Memoria que no puede ser direccionada directamente por el programador.
bus. *barra, conductor común.* Conexiones por donde las señales van desde unos orígenes a unos destinatarios.
Cada Unidad Central de Procesos (CPU) crea en el caso más general tres barras (buses).
byte. *octeto, byte.* Término que representa una porción medible de dígitos binarios consecutivos.

c

cache memory. *caché.* Memoria tampón de alta velocidad, utilizada entre el procesador central y la memoria principal.
call. *llamada.* Instrucción utilizada para transferir la secuencia de ejecución de programa a una subrutina o subprograma.

CAM. *memoria de contenido direccionable.* Memoria asociativa que es direccionada por el contenido en lugar de por la posición.
Captains. *Captains.* Acrónimo de character and pattern telephone access information network system; es un sistema viewdata utilizado en Japón.
card. *ficha.* Unidad de almacenamiento de información en forma de tarjeta, destinada a ser perforada según un código establecido de orificios, efectuados en posiciones codificadas, previamente definidas.
tarjeta. Unidad interna enchufable que contiene los conductores y componentes de los circuitos impresos.
card file. *archivo (fichero) de tarjetas.* Conjunto de tarjetas perforadas que guardan una información concreta, ordenadas con arreglo a una secuencia determinada con vistas a su utilización en una tarea específica.
carriage. *carro.* Dispositivo existente en determinados equipos impresores destinados a soportar y controlar el papel donde se imprime la información.
carryover. *acarreo.* Mandato que indica que ha de efectuarse un arrastre.
cassette. *cassette.* Funda plástica de pequeño tamaño que contiene y conserva las cintas magnéticas en orden a evitar la adherencia de partículas extrañas que podrían provocar errores o anomalías de lectura.
Ceefax. *Ceefax.* Servicio teletext utilizado por la BBC.
cerdip. *cerdip.* Contracción de la denominación en inglés de cápsula cerámica con patillas en disposición DIL.
chain. *cadena.* Secuencia cíclica de bits que forman las palabras de un código encadenado.
Dispositivo móvil empleado en las impresoras, que dispone, sobre unos eslabones colocados horizontalmente, de todos los caracteres existentes en el conjunto de impresión.
change record. *registro de modificación.* Registro cuya misión consiste en alterar o cambiar la información del registro principal correspondiente.
channel. *canal.* Conexión lógica entre una CPU y un dispositivo de E/S.
channel analog. *canal analógico.* Canal en el que la información transmitida puede adquirir cualquier valor dentro de los límites que previamente se han especificado para el mismo.
channel command. *mandato de canal.* Instrucción que indica a un canal, a una unidad de control o a un dispositivo una operación.
channel duplex. *canal dúplex.* Canal sobre el que se puede realizar una transmisión simultánea en ambas direcciones.
channel-four wire. *canal de cuatro hilos.* Circuito de doble vía sobre el que se transmiten señales simultáneamente por caminos separados y distintos, en direcciones opuestas, dentro del mismo medio de transmisión.
channel-half duplex. *canal semidúplex.* Canal sobre el cual se puede transmitir y recibir señales, pero solamente en una sola dirección a un tiempo.
channel-two wire. *par, canal de dos hilos.* Circuito formado por dos vías de transmisión, la cual puede efectuarse en cualquier dirección, pero no simultáneamente.
channel voice grade. *canal acústico.* Canal apto para realizar la transmisión de palabra hablada.
channel waiting queue. *cola de espera en canal.* Grupo de artículos que se han de tratar en un sistema que trabaja en tiempo real, los cuales permanecen a la espera de que el programa regulador de canal los tome en consideración.
character. *carácter.* Cada uno de los símbolos convencionales adoptados para representar, aisladamente o relacionados entre sí, la información. Representación efectiva o codificada de un dígito, letra o carácter especial.
character alignment. *alineación de caracteres.* Posición de los caracteres en relación al eje real de la línea de impresión.
character check. *comprobación de caracteres.* Comprobación que verifica la observancia de las reglas que determinan la formación de caracteres.
character error rate. *relación de caracteres erróneos.* Relación entre el número de caracteres recibidos correctamente y el número total de los transmitidos.
character generator. *generador de caracteres.* Circuito que forma las letras o números en un visualizador o impresora.
chart. *gráfico, diagrama.*
check. *verificación.* Dícese de cualquier verificación o reconocimiento realizado para contrastar la presencia de errores.
Error. Aviso que presenta la máquina al producirse un error.
check bit. *bit de verificación.* Bit asociado a un registro de información con el propósito de comprobar la ausencia de error en el mismo.
check out. *comprobación de resultados de salida.* Verificación de los resultados a la salida de un ordenador, localizando errores y efectuando las oportunas correcciones.
Verificar. Aplicar a un programa un método conducente a un diagnóstico.
checksum. *suma de verificación.* Resultado obtenido de la suma de los miembros que integran una serie de datos y que generalmente se añade al final de la misma, con la finalidad de detectar la presencia o ausencia de errores.
clear. *borrar, restaurar.* Acción de suprimir los datos de una memoria o de unas posiciones de memoria, sustituyéndolos por otros predeterminados.
clock. *reloj, generador de impulsos.* Dispositivo que genera una base de tiempos utilizada para proporcionar los impulsos secuenciales básicos para las operaciones de un ordenador, cuya forma de trabajo es secuencial.
CMOS. *MOS complementario.* Tecnología que se caractetiza por poseer muy bajo consumo. Esta tecnología requiere un canal p y un transistor de canal n, tiene la velocidad e integración intermedias entre NMOS y PMOS. Dispone de unas características ideales en cuanto a inmunidad al ruido.
CMRR. *relación del rechazo en modo común.* Ganancia de modo común en amplificadores operacionales.
COBOL. *lenguaje común para actividades comerciales.* Lenguaje de alto nivel simbólico de macroinstrucciones, destinadas al tratamiento de problemas esencialmente comerciales.
codec. *codificador-decodificador.*
coded decimal. *decimal codificado.* Tipo de notación en el que cada dígito decimal se identifica mediante un grupo de unos y ceros binarios.
combinational logic. *lógica combinacional.* Circuito que realiza funciones de lógica de Boole sin memoria.
common mode rejection ratio (CMRR). *relación del rechazo en modo común.* En amplificadores, término que se aplica para denotar el cociente entre la ganancia de tensión diferencial y la ganancia de tensión en modo común; por lo general se expresa en decibelios.
compiler. *compilador.* Programa de traducción que convierte instrucciones de alto nivel en un juego de instrucciones binarias (código objeto) para su ejecución.
computer. *ordenador, computador.* Sistema de cálculo de propósito general que incorpora una CPU, memoria, dispositivos de E/S, fuente de alimentación y un bastidor.
concentrator. *concentrador.* Dispositivo que posee diversos canales de entrada y un menor número de canales de salida. Equipo concebido para mejorar las prestaciones de los equipos de transmisión de datos que permite a los terminales o líneas compartir canales de transmisión.
console. *consola.* Terminal que dispone de la mayoría de las funciones de control en un sistema.
controller. *controlador.* Dispositivo encargado de dictar la secuencia de eventos en un sistema.
control unit. *unidad de control.* Módulo encargado de buscar y decodificar instrucciones. Esta unidad requiere un registro de instrucciones, un contador de programa y genera las señales de control y dirige el bus o barra de control.
conversational. *conversacional.* Sistema en el que el computador puede reaccionar después de cada sentencia del usuario o de entrada. También se denomina interactivo.
conversion rate. *velocidad de conversión.* Número de conversiones repetitivas A/D o D/A por segundo para un cambio a fondo de escala a una resolución y linealidad especificada.
conversion time. *tiempo de conversión.* Tiempo empleado por un convertidor analógico-digital para completar una sola conversión a una resolución y linealidad especificada para un cambio de entrada analógica a fondo de escala.
coprocessor. *coprocesador.* Dispositivo que realiza un tratamiento o proceso especializado en conjunción con el mi-

croprocesador principal de un sistema informático.

core. *núcleo, ferrita.* Pequeños toros magnéticos de ferrita empleados para almacenar información de un bit. Cilindro en cuyos laterales se incluyen unos rebordes con la finalidad de permitir arrollarse una cinta.

CPS. *caracteres por segundo.*

CPU. *unidad central de proceso.* Módulo encargado de buscar, decodificar y ejecutar instrucciones. Incorpora una unidad de control, una unidad aritmética y lógica (ALU) y dispositivos afines (registros, reloj, etc).

CPU time. *tiempo de la CPU.* Período de tiempo en el que la unidad central de proceso se dedica a la ejecución de instrucciones.

CROM. *memoria pasiva (ROM) de control.*

crossbar system. *sistema de barras cruzadas.* Sistema de conmutación de líneas que utiliza conmutadores de barras cruzadas.

crossfoot. *sumar y/o restar horizontalmente.* Dícese de la operación consistente en sumar varios campos horizontales de información numérica para verificar o totalizar.

crossprogram. *programa cruzado.* Programa para un ordenador A que reside (ejecuta) en un ordenador B.

crosstalk. *diafonía.* Interferencia entre dos circuitos.

CRT. *tubo de rayos catódicos.* TRC

crystal. *cristal.* Cristal de cuarzo cuya piezoelectricidad proporciona una frecuencia exacta para aplicar a la temporización del reloj. Se abrevia Xtal.

current loop. *lazo o bucle de corriente.* Medio de entrelazar vías de datos en presencia o ausencia de corriente en un cable bifiliar.

cursor. *cursor.* Punto móvil luminoso en forma de línea o subrayado mostrado en la pantalla de un TRC que indica el lugar de emplazamiento del carácter siguiente que ha sido introducido.

custom IC. *CI bajo pedido.* Circuito integrado fabricado según especificaciones del usuario.

cutoff. *corte.* Punto en el que la degradación que experimenta una señal, originada por la atenuación o la distorsión, hace que la señal no resulte utilizable.

CW. *palabra de control.* Palabra almacenada en memoria que se utiliza para controlar las prioridades que se han de observar en la selección de las operaciones a desarrollar cuando se emplean las técnicas de tiempo compartido.

cycle stealing. *robo de ciclos.* Método por el cual otra CPU, dispositivo de acceso a memoria u otro cualquiera gana acceso a las barras del microprocesador. En lugar de permitir el ciclo a la CPU para búsqueda y decodificación o ejecución, se utiliza por otro dispositivo que necesita utilizar las barras.

cycle time. *tiempo de ciclo.* Tiempo total empleado por un dispositivo (normalmente memoria) para completar su ciclo y estar disponible de nuevo. Normalmente, el tiempo de acceso es menor que el tiempo de ciclo y a veces igual.

cyclic redundancy check. *verificación por redundancia cíclica.* Método de verificación de errores que se emplea en la estación emisora y en la receptora cada vez que se acumula un carácter de verificación de bloques.

cyclic shift. *desplazamiento cíclico.* Desplazamiento en el que los datos que salen por un extremo del registro de memoria vuelven a entrar por el otro, como en un bucle cerrado.

cyclic store. *almacenamiento cíclico.* Tipo de memoria en el que el acceso a las posiciones de la misma se realizan periódicamente. Ejemplos clásicos de estas memorias son los tambores magnéticos.

d

D/A. *digital/analógico.* Conversión de una representación digital utilizada en ordenadores, para excitar altavoces, motores, etc. Por ello, los computadores pueden "hablar", "trabajar", etc. en el mundo real.

daisy chain. *cadena en margarita.* Método para priorizar interrupciones.

data acquisition. *adquisición de datos.* Obtención simultánea de datos a partir de sensores externos en forma analógica.

data acquisition system. *sistema de adquisición de datos.* Sistema que contiene una serie de circuitos que procesa una o varias señales analógicas y las convierte en digitales para su posterior proceso por un ordenador.

data area. *área de datos.* Porción de memoria utilizada durante la ejecución de un segmento.

data bank. *banco de datos.* Conjunto de información que puede ser procesable por un ordenador, independientemente de su medio de almacenamiento.

data carrier. *soporte de datos.* Medio físico sobre el que se registran los datos, de forma que, además de permitir su transporte, ha de permitir la interpretación de los mismos por los equipos correspondientes.

data cell. *celda de datos.* Volumen de almacenamiento de acceso directo que contiene bandas o láminas magnéticas en las que se registran y almacenan los datos.

data code. *código de datos.* Conjunto de normas por las que transmiten, reciben y procesan las señales que componen un bloque de información.

data collection. *recogida de datos.* Aplicación en la que la información se agrupa en un punto central antes de procesarla.

data communication. *comunicación de datos.* Proceso consistente en transmitir y recibir datos que comprende operaciones tales como codificación, decodificación, etc.

data link escape. *escape de enlaces de datos.* Carácter de escape utilizado para introducir la información de control en un flujo de datos.

data recovery filter. *Filtro de recuperación de datos.* Filtro utilizado para reconstruir una señal analógica a partir de un tren de muestras analógicas.

DAV. *datos disponibles.* Uno de los cinco bits de estado de un UART estándar.

D-bus. *barra D.* Barra de destino interno en una CPU que va de la ALU a los registros.

debbugging. *depuración, corrección, puesta a punto.* Proceso de exploración, localización y corrección de errores en la programación o en el funcionamiento de equipos.

decade counter. *contador de décadas.* Divisor por diez.

decay time. *tiempo de declinación.* Período de tiempo que tarda una tensión o corriente pulsante en disminuir hasta alcanzar un décimo de su valor máximo.

deceleration time. *tiempo de deceleración.* Tiempo transcurrido entre la lectura o escritura de un registro sobre una cinta magnética y el momento en que ésta se para.

decibel. *decibelio.* Unidad que expresa en forma logarítmica las relaciones de tensiones, corrientes o potencias.

decoder. *decodificador.* Unidad lógica que decodifica dos, tres, cuatro o más bits en salidas exclusivas mutuas.

dedicated register. *registro dedicado.* Registro empleado exclusivamente para contener un artículo específico.

delay line. *línea de retardo.* Circuito en cuyo diseño específico intervienen unos componentes que hacen posible la introducción de un retardo en la transmisión de la señal.

delta noise. *ruido delta.* Uno de los ruidos que se producen en el proceso de lectura de una célula magnética y cuya participación se hace patente sobre la tensión inducida del detector de conducción.

demodulation. *demodulación.* Proceso mediante el cual, se recupera información de una onda portadora modulada. Es un proceso inverso al de modulación.

determinism. *determinismo.* Grado al cual el movimiento de datos y direcciones pueden predecirse y seguirse cuando son operados por varios subsistemas microprocesadores.

demultiplexer. *demultiplexador.* Circuito lógico que puede dirigir una línea única de información digital a otras líneas.

development system. *sistema de desarrollo.* Sistema microcomputador que dispone de todos los elementos necesarios para el desarrollo de hardware y software de un microprocesador dado. Como mínimo incluye: μC, visualizador TRC (o teletipo), impresora, almacenamiento masivo, programador de PROM, lector de cinta de papel y emulador.

diagnostics. *diagnóstico.* Juego de rutinas empleadas para diagnosticar los fallos de funcionamiento de un sistema.

Didon. *Didon.* Sistema francés de teletext que utiliza las normas Antiope.

digitizer. *digitalizador.* Equipo convertidor encargado de transformar las señales analógicas en digitales.

direct access. *acceso directo.* Característi-

ca de ciertas memorias en las que los datos almacenados en ellas pueden serlo en una secuencia de direccionamiento que no guarda criterio alguno y cuyo acceso se puede realizar en forma directa, reduciéndose así el tiempo invertido.

direct addressing. *direccionamiento directo.* Técnica de direccionamiento empleada para instrucciones cortas, cuando el campo de dirección está limitado a 8 bits en lugar de a 16.

disable. *neutralización, deshabilitación.* Supresión o anulación de un dispositivo de interrupción.

diskette. *diskette, disco flexible.* Disco de mylar, recubierto de óxido magnético sobre el que se pueden almacenar datos por medio de una magnetización selectiva de posiciones en la superficie plana del mismo.

distributed system. *sistema distribuido.* Término que se aplica a una red de ordenadores en la cual existen ciertas funciones específicas que se reparten entre los diversos ordenadores de la red, de forma transparente al usuario.

DMA. *DMA.* Acceso directo a memoria. Técnica de entrada/salida para transferir datos entre un dispositivo y memoria sin ayuda del procesador de un ordenador.

DOS. *sistema operativo en discos.* Sistema operativo que integra funciones en fichero de disco, tales como: archivo simbólico, ubicación de espacio automática, y con frecuencia ubicación de memoria dinámica.

double-level multiplexing. *multiplexao de doble nivel.* Método de ampliación de canales en multiplexores analógicos por el que las salidas de un grupo de multiplexores se conecta a las entradas de otro multiplexor.

drum. *tambor.* Memoria magnética giratoria que utiliza la superficie de un cilindro.

dual-slope A/D conversion. *conversión A/D de doble pendiente.* Método indirecto de conversión A/D por el que se convierte una tensión analógica en un período de tiempo mediante un integrador y una referencia que posteriormente se mide por un reloj y un contador.

dump. *volcar.* Operación consistente en copiar el contenido total o parcial de una memoria. Este proceso, generalmente, es de memoria interna a memoria externa.

duplex. *dúplex.* Método de comunicación bidireccional que permite la transferencia simultánea de datos en ambas direcciones.

dynamic accuracy. *precisión dinámica.* Error total de un convertidor o sistema de conversión cuando trabaja a su máxima velocidad de conversión especificada.

dynamic memory. *memoria dinámica.* Memoria RAM MOS que utiliza circuitos dinámicos; cada bit se almacena en forma de carga en un transistor MOS.

e

EBCDIC. *código EBCDIC.* Código de 8 bits empleado por IBM para codificar símbolos alfanuméricos. Esencialmente es análogo al ASCII, pero con una codificación distinta.

echo. *eco.* Señal de conversación o de datos que se devuelve desde un punto distante con amplitud y retardo suficientes para molestar al que habla o interferir la transmisión normal de datos.

edge. *flanco.* Intervalo de tiempo durante el cual una señal lógica pasa del estado «0» al estado «1» o viceversa.

edit code. *código de edición.* Número o letra que indica que la edición ha de efectuarse según una pauta preestablecida.

editor. *editor.* Programa designado para introducción de texto en un sistema informático.

effective aperture delay. *retardo de apertura efectivo.* En un convertidor de muestreo y retención, es la diferencia de tiempos entre la orden de retención y el tiempo en el que la señal de entrada es igual a la tensión retenida.

EIA-RS232C. *EIA-RS232C.* Acoplamiento serie estándar para comunicaciones asíncronas.

electronic mail. *correo electrónico.* Servicio de mensajes mediante la utilización de equipos electrónicos y de telecomunicaciones que proporciona información escrita o visualizada.

emulation. *emulación.* Simulación en tiempo real. Un computador A emula a un computador B por ejecución de un emulador de forma que se comporta como el computador B.

enable. *habilitación, desinhibición.* Restauración al estado operativo de un dispositivo de desinhibición.

encode. *codificar.* Transformar la representación de una información por aplicación de un código, pero sin modificar el contenido de dicha información.

end. *final.* Sentencia que denota el final de un programa.

entry. *entrada.* Cada una de las instrucciones que generan una función dentro de un programa o de una subrutina. La primera instrucción de una subrutina es, forzosamente, una entrada, pudiendo existir dentro de la misma varias entradas más.

environment. *configuración.* Estado de todos los registros, ubicaciones de memoria y de otras condiciones operativas de un sistema.

EPROM. *memoria pasiva (ROM) programable.* Típicamente se refiere a las memorias PROM que pueden borrarse por exposición a los rayos ultravioleta.

erase. *borrar.* Concepto empleado para expresar la idea de suprimir unos datos almacenados sobre un soporte magnético, devolviendo cada uno de los elementos magnetizados a su estado nulo original, o incluso, a otro diferente al existente en ese momento.

error. *error.* Cualquier discrepancia entre los valores correctos y los obtenidos en algún tipo de cálculo o proceso.

error correcting code. *código corrector de errores.* Código generado para detectar errores y corregirlos. Para ello, lleva adicionados los elementos codificados necesarios para impedir, una vez detectado el error, hacer una corrección sobre la información original, sino sobre la equivocada, aprovechando la semejanza que en principio habrá entre ésta y la original.

even parity check. *comprobación de paridad par.* Verificación encaminada a averiguar si el número de dígitos 1 de un grupo de dígitos binarios es par o impar.

execute cycle. *ciclo de ejecución.* Tercer ciclo de los tres ciclos para ejecución de la instrucción de programa; durante este tiempo se lleva a cabo la operación de instrucción.

execution time. *tiempo de ejecución.* Tiempo requerido para la ejecución de una instrucción incluyendo la búsqueda, decodificación y ejecución.

expansion slots. *ranuras de ampliación.* Zócalos de la placa principal de un ordenador personal en los que se insertan placas accesorias.

extractor. *extractor.* Palabra situada en un registro que indica la parte de otra palabra con la que ha de llevarse a cabo la operación.



fallback. *situación de emergencia.* Condición que hace su aparición en el proceso, en la que se hace preciso emplear funciones especiales tanto de ordenador como manuales, o sustitutos totales o parciales de sistemas que funcionan defectuosamente.

fan-in. *cargabilidad de entrada.* Carga eléctrica presentada por una entrada a una salida.

fan-out. *cargabilidad de salida.* Carga eléctrica que puede excitar una salida. Normalmente se expresa como el número de entradas que pueden ser excitadas.

fetch. *búsqueda.* Toma que se hace en una memoria de un conjunto de información o de caracteres.

FIFO. *primero en entrar-primero en salir, cola de espera, FIFO.* Estructura en la cual los datos se depositan en un extremo y se extraen del otro. Las FIFO se utilizan como tampones para conectar dos dispositivos que trabajan en forma asíncrona y a distintas velocidades.

file. *fichero.* Bloque lógico de información, designado por un nombre y considerado como una unidad por el usuario. Un fichero puede estar dividido físicamente en registros, bloques o en otras unidades.

firmware. *soporte lógico inalterable, firmware.* Programa depositado en ROM.

flag. *señalizador.* Carácter que señala el acaecimiento de alguna condición, tal como el final de una palabra.

flag line. *línea de señalización.* Línea de señal utilizada en un enlace de datos para indicar la condición o estado de un dispositivo conectado a dicho enlace.

flash A/D converter. *convertidor A/D rápido, convertidor A/D en paralelo.* Dispositivo que trabaja según el método

miento en el que la dirección actual se obtiene por adición de un desplazamiento a la dirección de base.

indirect. *indirecto.* Modo de direccionamiento en el que la dirección de la ubicación en memoria a ser leída o escrita, está contenida en otro lugar de memoria o en un registro.

indirect type A/D converter. *convertidor A/D de tipo indirecto.* Término que se aplica a una variedad de convertidores analógico-digital que convierten la tensión de entrada desconocida en un período de tiempo y que posteriormente miden dicho período.

initialization. *inicialización.* Poner en funcionamiento un ordenador en un estado conocido.

input dynamic range. *margen dinámico de entrada.* Tensió pico a pico máxima permisible de un amplificador a través de los terminales de entrada que no origina límite o distorsión en la pendiente de salida.

inquiry. *consulta, interrogación.* Petición de información contenida en la memoria.

in-stream procedure. *procedimiento incorporado.* Conjunto de sentencias de control de trabajos que se colocan en la corriente de entrada y que pueden utilizarse un número indefinido de veces durante un trabajo.

instruction. *instrucción.* Sentencia única en un programa. Esta sentencia puede ser extraída de memoria, decodificada y ejecutada por la CPU. Pueden ser aritméticas, lógicas, de operación con registros, memoria y dispositivos de E/S u operaciones de control específicas.

instruction set. *juego de instrucciones.* Repertorio o colección de instrucciones que constituyen el lenguaje de programación de un ordenador o sistema de programación.

instruction time. *tiempo de instrucción.* Lapso de tiempo en el que se extrae una instrucción de la memoria principal de un ordenador.

instrumentation amplifier. *amplificador de instrumentación.* Circuito amplificador con entradas diferenciales de alta impedancia y elevado rechazo en modo común.

integrated-injection logic (I^2L). *lógica de inyección integrada, lógica I^2L.* Tecnología bipolar para fabricación de circuitos LSI. Se caracteriza fundamentalmente por su bajo consumo.

integrated software. *software integrado.* Paquete de software de aplicaciones que contiene programas que realizan más de una función.

integrating A/D converter. *convertidor A/D integrador.* Término aplicado a un tipo de convertidores A/D que basa su funcionamiento en la integración de la entrada analógica con el tiempo.

integrator. *integrador.* Circuito o dispositivo cuya función de salida es proporcional a la integral de la función de entrada respecto a una variable determinada.

interaction. *interacción.* En sistemas que trabajan en tiempo compartido, unidad de tipo básico que se emplea para medir la actividad del sistema.

interface. *acoplamiento, interconexión.* Conexión común a dos sistemas distintos de calculadores o a dos partes de un mismo sistema. Conceptos comunes que relacionan a dos equipos de diferentes funciones.

interface card. *placa de interconexión.* Tarjeta que convierte señales procedentes de un bus de ordenador en señales para un dispositivo periférico.

interlace. *entrelazar, concatenar.* Operación consistente en asignar los emplazamientos en memoria sobre un soporte móvil (tal como un tambor magnético).

interleaving. *interpolación.* Asignación de emplazamientos sucesivos de memoria a diferentes módulos físicos de memoria.

interpreter. *intérprete.* Programa de traducción empleado para ejecutar sentencias en lenguajes de alto nivel.

interrupt. *interrupción.* Señal de atención enviada por un dispositivo de E/S o chip a una MPU para romper el funcionamiento normal de la rutina que se está ejecutando. Señal que origina la pausa.

I/O. *entrada/salida, E/S.* Líneas o dispositivos empleados para obtener o visualizar la información en el exterior.

iteration. *iteracción.* Repetición de un grupo de instrucciones.

j

job. *trabajo.* Unidad de código que resuelve un problema, un programa y todas las subrutinas y datos relativos a ella.

joggle. *empujar, emparejar tarjetas.* Operación consistente en agitar con la mano un fichero de tarjetas perforadas para introducirlas alineadas en el almacén de alimentación de la máquina que ha de trabajar con ellas. Proceso de colocar parejas de tarjetas con arreglo a un criterio determinado.

joystick. *palanca de mando.* Palanca normalmente de tipo vertical que puede inclinarse en cualquier sentido para indicar dirección o movimiento. Se utiliza para desplazar un punto en pantalla.

jump. *salto.* Desviación con respecto a la secuencia normal de ejecución de las instrucciones en un ordenador.

juxtaposition. *yuxtaposición.* Acción de colocar entidades, elementos o unidades de modo que quedan adyacentes o contiguos.

k

K. *K.* Símbolo equivalente a 10^3.

key. *clave, tecla.* Conjunto de caracteres de un registro o de un conjunto de registros que se utilizan para identificar a éstos. Cualquier tecla, conmutador, etc., que puede cerrar o abrir un circuito o generar un movimiento mecánico.

keyboard. *teclado.* Grupo de teclas que se utiliza para introducir información en un sistema ordenador.

keyword. *palabra clave.* Palabra característica de un fichero, que se utiliza para sacar su contenido mediante significado.

KIPS. *KIPS.* Mil instrucciones por segundo.

l

label. *rótulo, etiqueta.* Conjunto de caracteres de un registro colocados en el mismo para su identificación.
Identificación visible que explica el contenido de un almacenamiento externo.

lace. *perforación en cadena.* Operación consistente en perforar todas las posiciones de una columna en una ficha.

lag. *retardo.* Intervalo de tiempo existente entre dos estados.

language. *lenguaje.* Conjunto de reglas y símbolos estructurados de forma que la combinación de éstos ofrece un significado específico de comunicación.

language algorithmic. *lenguaje algorítmico, ALGOL.* Lenguaje simbólico de carácter científico.

language character set. *juego de caracteres del lenguaje.* Conjunto de caracteres definidos para utilizar en un lenguaje de programación.

language object. *lenguaje objeto, lenguaje máquina.* Lenguaje comprensible por la máquina, el cual resulta generalmente de la traducción de un lenguaje simbólico. Está formado por una serie secuencial de instrucciones en lenguaje máquina.

language translator. *traductor de lenguaje.* Vocablo de tipo genérico con el que se define cualquier ensamblador, compilador u otro tipo de rutina, que admite sentencias en un lenguaje de programación y produce sentencias equivalentes en otro, sin cambiar de forma apreciable su significado.

last-in-first-out (LIFO). *último en entrar-primero en salir, pila, (LIFO).* Término que describe un método de extracción de elementos o artículos de una lista ordenada en cola.

latch. *cerrojo.* Dispositivo físico que captura la información y la retiene.

latency. *espera.* En dispositivos de almacenamiento rotativos, dícese del retardo transcurrido entre el instante en que el dispositivo es notificado de la llegada de una transferencia y el instante en que el dispositivo está listo para realizarla.

learning curve. *curva de aprendizaje.* Mejoras en el proceso de fabricación respecto a la experiencia.

least significant bit (LSB). *bit menos significativo.* Dígito binario que tiene un peso de 2º ó 1.

library. *biblioteca.* Colección de programas.

light pen. *lápiz fotosensible.* Dispositivo de entrada para un TRC que registra la

emisión de luz en el punto de contacto con la pantalla. La relación de temporización con el comienzo de un barrido, determina la posición aproximada en la pantalla.

linearity error. *error de linealidad.* Desviación máxima de la función de transferencia de un convertidor de datos desde la línea recta ideal con errores de ganancia y desplazamiento a cero. Por lo general se expresa en bits menos significativos o en porcentaje del límite de escala completa.

line delay. *línea de retardo.* Dispositivo concebido para retardar la transmisión de una información determinada.

line printer. *impresora de líneas.* Impresora de alta velocidad capacitada para imprimir de forma simultánea una línea completa.

link. *enlace.* Indicador de dirección al próximo elemento de una lista o de un bloque contiguo de un fichero.

linkage. *enlace.* En programación, dícese del código que conecta dos rutinas codificadas por separado y que pasa los valores y/o el control entre ellas.

listing. *listado.* Documento listado, generalmente sobre formularios continuos de la información obtenida por el ordenador.

loader. *cargador.* Programa especial que toma la información en formato binario y la pone en memoria.

local area network. *red local.* Red de alta velocidad que conecta ordenadores, impresora y otros periféricos en una oficina o edificio.

location. *ubicación, emplazamiento.* Lugar numerado o denominado en memoria en el que una unidad de datos o una instrucción es almacenada.

lockout. *bloqueo.* Proceso que automáticamente se aplica en un sistema, mediante el cual una o varias partes del mismo se inhiben del resto de las operaciones durante el tiempo en que éstas no deben participar en ellas.

logger. *registrador.* Dispositivo que registra automáticamente las condiciones físicas respecto al tiempo.

log-on. *identificarse.* Acto de presentación o autoidentificación por el que un usuario se hace reconocer por el sistema.

long term stability. *estabilidad a largo plazo.* Variación de la precisión de un convertidor de datos debido exclusivamente a cambios de tiempo. Por lo general se expresa en porcentaje por mil horas o por año.

look-up table. *tabla de consulta.* Colección de datos dispuestos en forma adecuada, tendentes a una fácil consulta, almacenados en posiciones de máquina ordenadas en secuencia, para mayor facilidad de anotación.

loop. *bucle.* Circuito físico por el que es factible realizar una transmisión.

low order. *orden inferior.* Relativo a la importancia o significado atribuido al dígito situado en el extremo derecha de un número expresado en notación posicional.

LPM. *líneas por minuto.* Unidad que determina la velocidad de impresoras.



m

machine. *máquina.* Dispositivo o equipo capaz de realizar por si solo alguna función.

machine code. *código de máquina.* Sistema de codificación adoptado en el diseño de un ordenador para representar su juego de instrucciones.

machine error. *error de máquina.* Error introducido en los resultados de un proceso automático que puede atribuirse al funcionamiento defectuoso de una máquina, más que a un fallo en el manejo del equipo.

machine instruction. *instrucción de máquina.* Instrucción escrita en código de máquina de un ordenador; es decir, aquella que la máquina puede realizar directamente sin necesidad de traducción.

machine language. *lenguaje de máquina.* Instrucciones escritas en código de máquina que un ordenador puede obedecer directamente sin necesidad de traducción.

macro instruction. *macroinstrucción.* Instrucción individual que se escribe como parte de un lenguaje fuente y que cuando se compila en un programa en código de máquina, genera varias instrucciones en código de máquina.

magnetic core storage. *almacenamiento de núcleo magnético.* Conjunto de núcleos magnéticos dispuestos en matrices para formar la memoria de un ordenador.

magnetic disk. *disco magnético.* Dispositivo de almacenamiento que consta de varios platos circulares planos que están revestidos en ambas caras por una capa de material magnetizable.

mailbox. *buzón.* Lugar en el que una tarea puede enviar un mensaje para su extracción por otra tarea.

main frame. *unidad central.* En un principio comprendía la estructura principal de una CPU, en la que estaban instalados la unidad aritmética y la circuitería lógica asociada. Actualmente se refiere, en sentido coloquial, al propio procesador central.

main memory. *memoria principal.* Memoria interna de un ordenador; es decir, almacenamiento de acceso inmediato, a diferencia de cualquier almacenamiento auxiliar que pueda formar parte del sistema del ordenador.

main program. *programa principal.* Parte central de un programa que generalmente transfiere el control a otras subrutinas, según la naturaleza de los datos que se están procesando. Estructura central en la que se ensamblan las diversas secciones o subrutinas.

main storage. *almacenamiento principal.* Almacenamiento desde el cual se ejecutan las instrucciones. Por lo general, es el almacenamiento más rápido de un ordenador.

mark. *marca.* Carácter empleado para identificar el final de un conjunto de datos.

masking. *enmascaramiento.* Técnica que consiste en utilizar una máscara para operar sobre la configuración de bits de algún otro operando, en orden a alterar o aislar ciertas posiciones de bit.

mass storage. *almacenamiento masivo.* Sistema de almacenamiento auxiliar de gran capacidad que está conectado directamente con una CPU.

matching. *comparación con selección.* Técnica que consiste en comparar las claves de dos registros para seleccionar artículos con destino a una fase particular del proceso o para rechazar los registros inválidos.

matrix store. *memoria matricial.* Memoria cuyos elementos están dispuestos en forma tal que el acceso a cualquier posición exige el empleo de dos o más coordenadas.

mega. *mega.* Prefijo que indica 10^6.

memory. *memoria.* Dispositivo en el que se pueden introducir datos, donde se mantienen y donde pueden retirarse posteriormente. En sentido genérico, dícese de cualquier dispositivo capaz de almacenar datos.

memory dump. *vuelco o vaciado de memoria.* Proceso de transferencia del contenido, total o parcial, de una memoria a otra cualquiera, generalmente para comprobar el contenido de la misma.

memory fill. *carga de memoria.* Técnica empleada para depuración de programas, mediante la cual se llena la memoria libre de unos caracteres que impedirán la ejecución del proceso si éste intenta ejecutar instrucciones en dichas áreas.

memory guard. *protección de memoria.* Dispositivo que forma parte de los componentes físicos del ordenador o de su soporte lógico, que tiene por función impedir que un programa direccione posiciones específicas de la memoria interna.

menu. *menú.* Lista de órdenes opcionales disponible para el usuario de un programa de ordenador.

mercury memory. *memoria de mercurio.* Dispositivo de almacenamiento en el que la información se retiene haciendo recircular señales en una línea de retardo de mercurio.

merge. *fusión.* Proceso que se utiliza para formar una secuencia clasificada de registros, partiendo de dos o más secuencias previamente clasificadas.

micro. *micro.* Prefijo que indica 10^{-6}.

microinstruction. *microinstrucción.* Instrucción que forma parte de una sección de microcodificación. Uno de los pasos constitutivos de una macroinstrucción.

milli. *mili.* Prefijo que indica 10^{-3}.

misfeed. *pérdida de alimentación.* Funcionamiento defectuoso de un equipo cuyo sistema de alimentación de información ha de efectuarse mediante tarjetas perforadas, por mala perforación, mala lectura o deterioro de la tarjeta.

mistake. *error, equivocación.* Fallo, cualquiera que sea la circunstancia, imputable a la responsabilidad humana.

modem. *modulador/demodulador, modem.* Dispositivo que permite efectuar la transmisión de datos a grandes distancias sin errores.

monitor. *monitor.* Cualquier dispositivo que examina el estado de un sistema para indicar cualquier desviación que se produzca con respecto a las condiciones de funcionamiento.
monotonicity. *monotonicidad.* Característica de un convertidor /D relativa a la función de transferencia por la que un código de entrada en aumento produce un incremento continuo de la salida analógica.
most-significant-character (MSC). *carácter más significativo.* En la representación posicional, dícese del carácter situado en la posición extremo izquierda de un grupo de caracteres significativos.
MS-DOS. *MS-DOS.* Microsoft Disk Operating System. Sistema operativo del ordenador IBM PC y máquinas con él compatibles.
multi-access. *acceso múltiple, multiacceso.* Un sistema de acceso múltiple es aquel que permite a diversas personas realizar las funciones interactivas, generalmente relacionadas con la consola del operador.
multiple-address. *dirección múltiple.* Término relativo a las instrucciones que especifican la dirección de más de una posición de memoria.
multiplexer. *multiplexor, multiplexador.* Dispositivo de control de comunicaciones que permite conectar una unidad central de proceso a un gran número de canales de comunicaciones distintos, cualquiera de los cuales (o la totalidad) puede estar transfiriendo datos desde/o a la unidad de proceso.
multiplying D/A converter. *convertidor D/A multiplicador.* Convertidor en el que la tensión de referencia puede variarse en una amplia gama para producir una salida analógica que es producto del código de entrada y de la tensión de referencia de entrada.
multiprocessing. *multiproceso.* Proceso mediante el cual se emplean varios calculadores para el tratamiento de una sola transacción.
multiprocessor. *multiprocesador.* Procesador central que contiene dos o más unidades aritméticas independientes además de su correspondiente lógica de control.
multiprogrammation. *multiprogramación.* Técnica que permite tratar simultáneamente varios programas, o fragmentos de programas, diferentes, ejecutándose la intercalación de los mismos bajo el control de un programa supervisor.
multisequential system. *sistema multisecuencial.* Sistema que puede intercalar instrucciones pertenecientes a programas o secuencias diferentes, ejecutándolas de una sola vez.
multistation. *circuito multiterminal.* Circuito que interconecta varios terminales situados en distintas posiciones, haciendo llegar la información transmitida simultáneamente a todos ellos.
multitasking system. *sistema multitarea.* Sistema operativo que puede pasar dos o más programas o tareas concurrentemente.

n

NAND. *NO-Y.* Puerta NO-Y. (Véase NOT-AND).
nanosecond. *nanosegundo.* Prefijo que denota la mil millonésima de segundo (1×10^{-9} segundos).
natural function generator. *generador de función natural.* Generador en el que la función se basa o formula en una ley física o matemática.
natural language. *lenguaje natural.* Lenguaje cuyas reglas reflejan y describen el uso común, más que el uso prescrito.
n-core per-bit store. *memoria de n núcleos por bit.* Memoria en la que cada unidad elemental de información magnética está integrada por n núcleos magnéticos.
nesting. *jerarquización.* Término relativo a una rutina o subrutina que contiene una estructura similar a sí misma.
network. *red.* Relativo a cualquier sistema, serie de puntos y sus interconexiones.
no-address instruction. *instrucción sin dirección.* Instrucción que no necesita especificar una dirección en la memoria.
noise. *ruido.* Señales extrañas, generalmente no deseadas que surgen circunstancialmente en cualquier parte de un sistema de transmisión de datos.
noise rejection. *rechazo de ruido.* Cantidad de supresión del ruido de entrada analógico de modo normal de un circuito; por lo general se expresa en decibelios.
n+one address instruction. *instrucción de n+una dirección.* Formato de instrucción que incluye n+1 direcciones, siendo, por lo general, la dirección n+1 la que indica la instrucción que ha de ejecutarse a continuación.
non-return-to-zero recording. *registro de información sin retorno a cero.* Método de registrar magnéticamente bits, en el que la magnetización empleada para representar ceros y unos ocupa totalmente el elemento de almacenamiento y en el que no es posible volver a una condición de referencia entre bits.
non-volatile memory. *memoria no volátil, memoria estable.* Medio de almacenamiento cuyo contenido no se mutila cuando se corta la corriente y se encuentra disponible cuando retorna la alimentación.
NOR-gate. *puerta NO-O.* Elemento lógico en el que la variable representada en la única señal binaria de salida es la no disyunción de las variables representadas en las señales binarias de entrada.
normal-mode rejection. *rechazo en modo normal.* Atenuación de una frecuencia específica o banda de frecuencias que aparece directamente a través de dos terminales.
NOT-AND. *NO-Y.* Operación lógica que tiene la propiedad de que, si p es una expresión, q es una expresión y r es una expresión, entonces p, q, r,.... son verdaderas si por lo menos una de las expresiones es falsa; y p, q, r,.... son falsas si todas las expresiones son verdaderas.
null instruction. *instrucción nula.* Sinónimo de no operación.
number cruncher. *trituranúmeros.* Término de la jerga que se aplica a los ordenadores de gran potencia de cálculo, cuya característica más destacable es su capacidad para manejar cifras grandes.
numeric character. *carácter numérico.* Cualquier carácter empleado como dígito en la representación de números.
numeric coding. *codificación numérica.* Dícese de cualquier sistema de codificación que únicamente utilice números.
numeric word. *palabra numérica.* Palabra formada por dígitos y, posiblemente, por caracteres de espaciado y especiales.

o

object code. *código objeto.* Salida de un compilador o de un ensamblador, que por su propia naturaleza, es un código ejecutable de máquina o resulta apropiada para que el montador la procese con el fin de producir un código ejecutable de máquina.
object configuration. *configuración objeto.* Conjunto de equipos de un ordenador encargado de realizar el programa objeto.
object language. *lenguaje objeto.* Lenguaje o juego de instrucciones codificadas al cual se traduce un lenguaje fuente, valiéndose de un compilador.
object program. *programa objeto.* Programa obtenido en lenguaje de máquina después de la compilación o ensamblaje de otro simbólico. Es, pues, un programa simbólico traducido a otro de máquina.
object routine. *rutina objeto.* Rutina en lenguaje de máquina que constituye la salida de la traducción del lenguaje fuente. Rutina de ejecución.
OCR (optical character recognition) *reconocimiento de caracteres ópticos.* Reconocimiento por medios automáticos de caracteres impresos por procedimientos fotoeléctricos.
octal. *octal.* Término relativo al sistema de representación numérica con base ocho.
octet. *octeto.* Carácter o posición compuesta de ocho bits.
odd-even check. *control de paridad par-impar.* Forma de verificación de paridad en la que se agrega un bit adicional o una palabra o carácter; este bit tiene el valor 1 o 0, dependiendo de que el número de bits 1 de la palabra o carácter sea par o impar.
odd-parity check. *comprobación de paridad impar.* Verificación de paridad en el que se espera que el número de unos o ceros de un grupo de dígitos binarios sea impar; incluyendo en la cuenta el propio bit de paridad.
off-line. *fuera de línea, autónomo.* Se dice

que una parte de un sistema de ordenador está fuera de línea cuando no se halla bajo el control directo de la CPU.

off-line unit. *dispositivo autónomo.* Dispositivo E/S o equipo auxiliar que no está bajo control directo de la CPU.

offset. *desfase.* Diferencia entre el valor o condición buscada y el que realmente se obtiene.

offset drift. *deriva de desplazamiento.* Cambio del cero analógico con la temperatura en un convertidor de datos que trabaja en modo bipolar. Por lo general se expresa en partes por millón/°C del FSR.

offset error. *error de desplazamiento.* Error a cero analógico en un convertidor que trabaja en modo bipolar.

off-time. *tiempo de inactividad.* Tiempo que transcurre cuando un ordenador no está programado para su uso, mantenimiento, reparación, o modificación.

on-line. *en línea, en conexión directa.* Se dice que una parte de un sistema de ordenador está en línea, cuando se halla bajo el control directo de la CPU.

open architecture. *arquitectura abierta.* Diseño de sistema operativo o de ordenador cuyas especificaciones son facilitadas y hechas públicas por el fabricante, tendente a que otras firmas produzcan circuitos o programas compatibles.

open loop. *bucle abierto.* Sistema de control, en el que la acción correctora no es automática, sino que depende de una intervención externa determinada por la información que aparece representada.

operand. *operando.* Elemento o dato de una operación del cual se obtiene el resultado por medio de acciones definidas.

operating system. *sistema operativo.* Conjunto de rutinas encargadas de la supervisión de las operaciones de un sistema ordenador (carga, situación en memoria, sucesión de operaciones, ensemblaje, compilación, etc.). El sistema operativo, en cada caso, introduce automáticamente el programa apropiado.

operating system shell. *núcleo de sistema operativo.* Software que traduce instrucciones introducidas por el usuario de ordenador en órdenes al sistema operativo.

operation. *operación.* Acción determinada por medio de la cual se obtiene un resultado de un operando. Acción definida por una sola instrucción de ordenador. Acción definida por un solo elemento lógico.

operation register. *registro de operaciones.* Registro en el que se almacena el código de operación durante el ciclo de operación.

optical-bar code reader. *lector óptico de códigos de barras.* Dispositivo que lee por medios ópticos la información codificada mediante marcas efectuadas en los documentos con lápiz tinta u otros medios.

Oracle. *Oracle.* Servicio de teletex utilizado en Gran Bretaña por la IRA.

ordering. *ordenación.* Clasificación o disposición en secuencia o en serie.

OR-gate. *puerta O.* Elemento lógico en el que la variable representada en la única señal binaria de salida es la disyunción de las variables representadas en las dos o más señales binarias de entrada de que dispone el elemento. Es decir, la señal binaria de salida será 1 cuando una o más señales binarias de entrada sean 1.

outboard recorder. *registrador externo.* Rutina o módulo que registra los datos pertinentes en el fichero registrador del sistema cuando se produce un error de E/S no recuperable.

output. *salida.* Resultados producidos por un ordenador. Transferencia de información desde una CPU a un dispositivo de salida.

output queue. *cola de salida.* Término colectivo con el que se designan las 36 colas de control que describen conjuntos de datos de salida del sistema.

output stream. *corriente de salida.* Mensajes de diagnósticos y otros datos que el sistema de explotación o un programa de proceso emiten en dispositivos de salida especialmente activados para este fin.

output unit. *unidad de salida.* Dispositivo o dispositivos que se integran en la configuración de un equipo de tratamiento automático de la información, cuya única finalidad es proporcionar la salida al proceso en curso.

overflow. *rebasamiento.* Efecto producido cuando el resultado de una operación aritmética excede la capacidad de valores admitidos para la representación de un número o de un registro.

overlapping. *solapamiento.* Término que designa a toda técnica o dispositivo que permite realizar varias informaciones al mismo tiempo.

overlay. *recubrimiento.* Técnica utilizada cuando las necesidades de memoria son superiores a las disponibilidades. Consiste en realizar la carga de rutinas en memorias de alta velocidad procedentes de otras, de forma que las mismas posiciones de memoria se ocupen por diferentes rutinas en momentos distintos.

overload. *sobrecarga.* Exceso de mensajes que se producen durante un momento determinado en las líneas de comunicación debidos a la transmisión simultánea de datos a un ordenador.

overwrite. *sobregrabar.* Técnica consistente en grabar información en una posición de memoria, destruyendo la información que contenía anteriormente.

P

pack. *lote, fichero de tarjetas.* Colección de tarjetas perforadas relativas a un mismo asunto y que generalmente se ordenan siguiendo un criterio específico. *Empaquetar, condensar.* Acción de situar más de una unidad de información en una sola unidad de almacenamiento en orden a un ahorro de espacio de memoria.

padding. *relleno.* Acción de agregar blancos o caracteres no significativos al final de un registro o bloque, con objeto de darle un tamaño determinado.

page-at-a-time printer. *impresora de páginas completas.* Tipo de impresora en el que la impresión puede efectuarse por páginas completas, si bien el formato de los caracteres de dicha página ha de ser definido con anterioridad.

paper tape. *cinta de papel perforado.* Cinta de papel de tipo apergaminado destinada a recibir unas perforaciones realizadas en posiciones específicas que representan la información.

paper throw. *salto de papel.* Movimiento del papel por la impresora que difiere del avance normal del papel para su impresión y se efectúa a una velocidad mayor que la del espaciado de una sola línea.

parallel. *paralelo, en paralelo.* Método de tratamiento simultáneo de todos los elementos de unidad de información.

parallelism. *paralelismo.* Ejecución concurrente de más de una función en un microprocesador.

parallel D/A converter. *convertidor D/A en paralelo.* Tipo de convertidor en el cual, bajo la aplicación de un código de entrada, todos los bits cambian de forma simultánea tendente a producir una nueva salida.

parallel storage. *almacenamiento en paralelo.* Sistema de almacenamiento en el que el tiempo que se necesita para acceder a una zona situada en el almacenamiento es el mismo que el tiempo necesario para acceder a cualquier otra zona del almacenamiento.

parallel transfer. *transferencia en paralelo.* Método de transferencia de datos en el que la transmisión de éstos se realiza de forma simultánea.

parameter. *parámetro.* Valor fijo que se concede a una variable, durante el proceso de un problema específico por medios automáticos.

parity bit. *bit de paridad.* Bit de verificación o su complemento que se añade a los bits de un bloque de información para controlar la transferencia de dicho bloque entre las unidades del equipo, pero que no tiene valor como tal información.

parity check. *verificación de paridad.* Comprobación que se lleva a cabo durante la transferencia de datos. Consiste en sumar los bits de una unidad de datos, calcular el bit de paridad necesario y comprobar el bit de paridad calculado, comparándolo con el bit de paridad que se transfiere con la partida de datos.

pass. *pasada.* Paso de una cinta magnética por las cabezas de lectura.

patch. *parche.* Grupo de instrucciones que se agregan a una rutina para corregir una equivocación.

path. *vía, camino.* Secuencia de instrucciones de un programa de ordenador.

peripheral control unit. *unidad de control de periférico.* Unidad encargada de controlar las operaciones de una o más unidades periféricas pertenecientes a un sistema de tratamiento automático de datos y que permanecen bajo el control de la CPU de dicho sistema.

peripheral processor. *procesador periférico.* En un sistema de proceso de datos en el que se emplea más de una unidad de proceso, se denomina procesador periférico a aquél que opera bajo el control de otro.
peripheral unit. *periférico.* Máquina que puede funcionar bajo el control del ordenador. El equipo periférico consta de dispositivos de E/S y de almacenamiento.
picosecond. *picosegundo.* Milésima parte de un nanosegundo. 10^{-12} segundos.
pinboard. *tablero de conexiones.* Cuadro de control en el que las interconexiones se realizan mediante conectores desprovistos de cable y en forma de clavija.
pipelining. *segmentación.* Preparación simultánea para la ejecución de una o más instrucciones mientras otra se está ejecutando.
Pixel. *Pixel.* Acrónimo de picture element. Término aplicado al elemento más pequeño controlable que puede representarse en una pantalla de visualización.
PL/I. *lenguaje PL/I.* Lenguaje de programación simbólico que combina las características típicas de los lenguajes administrativos con las de los lenguajes científicos.
plotter. *trazador de gráficos.* Dispositivo que delinea o traza una imagen visual de una función o tabla.
plug-in unit. *unidad enchufable.* Conjunto de componentes electrónicos normalizados unidos entre sí por conductores, que puede enchufarse y desenchufarse fácilmente.
point. *coma, punto.* Símbolo que separa la parte entera de la fraccionaria de un número.
pointer. *indicador.* Elemento de programa que hace referencia a otro.
point-to-point transmission. *transmisión punto a punto.* Transmisión que se realiza directamente entre dos puntos sin intervención de terminal intermedio ni ordenador.
polling. *sondeo.* Técnica empleada para controlar las líneas de comunicación. Para ello, el dispositivo de control emite una señal a cada uno de los terminales conectados, estableciendo comunicación cuando el interrogado tiene mensaje para transmitir.
port. *vía de acceso.* Parte de un procesador de datos que se dedica a un solo canal de datos con el fin de recibir o transmitir éstos a uno o más dispositivos externos situados a distancia.
pre-edit. *preeditar.* Técnica consistente en realizar una pasada preliminar de edición de los datos de entrada, antes de emplear los datos para un proceso posterior.
Prestel. *Prestel.* Servicio públicovideotex utilizado por British Telecommunications.
printout. *impresión de salida.* Término general con el que se designa la salida de una impresora; páginas impresas producidas por una impresora.
private database. *base de datos privada.* Base de datos que contiene datos restringidos que es únicamente accesible a un grupo seleccionado de usuarios.
problem-oriented language. *lenguaje orientado a un tipo de problemas.* Lenguaje especialmente diseñado para su aplicación a un tipo específico de problemas y cuya utilidad en cualquier problema de otras características es prácticamente nula.
process. *proceso.* Término que indica la ejecución de todas las operaciones de un tratamiento: lectura, compilación, cálculo, etc.
processing unit. *unidad de proceso.* Parte del ordenador que proporciona un área de almacenamiento para programas y datos y efectúa las operaciones especificadas en el programa.
processor. *procesador, unidad de proceso.* Término general que designa cualquier dispositivo capaz de desarrollar operaciones con datos.
program/programme. *programa.* Grupo de instrucciones compuestas con la finalidad de resolver un problema específico mediante un ordenador.
programming. *programación.* Proceso que comprende el diseño, escritura y prueba de un programa. Arte de reducir a instrucciones detectables por la máquina el plan para la resolución de un problema.
program testing time. *tiempo de prueba del programa.* Tiempo invertido en la comprobación de un programa de ordenador.
propagation delay. *retardo de propagación.* Tiempo total invertido por una señal en su desplazamiento de un punto a otro de un circuito.
protocol. *protocolo.* Conjunto de normas que gobiernan la forma de comunicación entre dos equipos.
pseudo-instruction. *pseudoinstrucción, instrucción falsa.* Grupo de caracteres que tienen el mismo formato general que una instrucción de ordenador, pero que éste nunca ejecuta como una instrucción real.
punch. *perforar.* Acción consistente en efectuar un orificio en una tarjeta perforada o en cinta de papel para su uso.
push down store. *memoria de almacenamiento descendente.* Memoria que constituye, en cuanto a los componentes de la máquina, la aplicación o representación práctica de una lista de desplazamiento descendente. Contrasta con el término "Push up store".

q

quantizer. *cuantificador.* Dispositivo que convierte una cantidad analógica a su equivalente digital.
quantizing error. *eror de cuantificación.* Incertidumbre inherente en la digitalización de un valor analógico a causa de la resolución finita del proceso de conversión.
quantum. *cuanto.* Cantidad elemental indivisible, por la cual se puede variar una magnitud física determinada.
queue. *cola de espera.* Grupo de elementos de un sistema que espera su tratamiento.
quotient. *cociente.* Resultado de dividir un operando, denominado dividendo, por otro llamado divisor, denominándose resto al excedente que se produce cuando el dividendo no es múltiplo del divisor.

r

radial transfer. *transferencia radial.* Transferencia de datos entre equipos periféricos y la memoria principal.
radix. *raíz, base.* Número entero adoptado para definir un sistema de numeración.
radix complement. *complemento de la base.* Número obtenido al restar cada dígito de una cantidad a la base menos uno, añadiendo después 1 al dígito menos significativo, efectuando el arrastre de números necesarios.
random-access memory (RAM). *memoria de acceso aleatorio, memoria RAM.* Almacenamiento concebido para proporcionar un tiempo de acceso constante para cualquier posición direccionada, cualquiera que sea la posición previamente direccionada.
random-number sequence. *secuencia de números aleatorios.* Serie de números imprevisibles producidos por cambios, que satisface una o más de las pruebas de aleatoriedad.
range. *rango, escala.* Conjunto de valores que puede tomar una cantidad.
rank. *jerarquizar.* Acción de disponer series según su importancia.
raw data. *datos en bruto.* Datos que no han sido procesados por ningún sistema.
read. *leer.* Acción de obtener datos almacenados en memoria.
read-only storage (ROM). *memoria de sólo lectura, memoria pasiva, memoria inalterable, memoria ROM.* Dispositivo capaz de retener datos, los cuales no se pueden alterar por instrucciones de programa.
read-out. *lectura de salida.* Lectura realizada por la memoria interna y transferida a un sistema de almacenamiento de tipo externo.
read time. *tiempo de lectura.* Tiempo empleado en localizar datos en una sección de memoria y transferirlos a una unidad aritmética.
real-time. *tiempo real.* Término que se aplica al tiempo actual, durante el cual transcurre el proceso. Método de proceso de datos realizado a tal velocidad que virtualmente no transcurre tiempo alguno entre el momento que se formula la consulta y el instante en que se recibe el resultado.
record. *registro.* Grupo de elementos de datos relacionados entre sí que son tratados como una sola unidad.
redundancy. *redundancia.* Empleo de caracteres o bits adicionales que se agregan a un grupo de datos a fin de proporcionar un medio de comprobación de la exactitud de los datos.
redundant code. *código de redundancia.*

Código que utiliza más elementos de señal que los estrictamente necesarios para representar la información esencial para el proceso o a la que se ha de transmitir.

reel. *rollo, bobina.* Bobinado de cinta de papel, montado en un soporte de cartón o plástico.

re-entrant code. *código reentrante.* Bloque de instrucciones que pueden ejecutarse por dos o más tareas concurrentemente sin interferencia.

reference data base. *base de datos de referencia.* Base de datos que en principio sirve como un puntero a fuentes originales de información.

reference time. *tiempo de referencia.* Tiempo empleado por un impulso para iniciar una acción y que tarda en alcanzar el diez por ciento de su amplitud especificada.

register. *registro.* Números de posiciones de memoria que contienen una información referida a un aspecto específico. Dispositivo hardware, cuya función consiste en retener una información que se ha de tratar a continuación.

relative address. *dirección relativa.* Dirección especificada en una instrucción que no ha sido modificada.

relative code. *código relativo.* Código de programa en el que las direcciones se especifican en relación con alguna dirección de base o código en el que se emplean direcciones simbólicas.

relative coding. *codificación relativa.* Escritura de instrucciones de programa que se lleva a cabo utilizando las técnicas del direccionamiento relativo.

relative error. *error relativo.* Relación de un error existente en algún resultado calculando respecto al valor cuantitativo del resultado.

relocate. *reubicar.* Acto de modificar automáticamente las instrucciones de un programa, que se lleva a cabo para permitir la carga y ejecución del programa en cualquier área de memoria.

remainder. *resto.* Diferencia entre el dividendo de una división y el producto del divisor por el cociente.

repertoire/repertory. *repertorio, juego.* Gama de caracteres o de códigos individuales de que se dispone en un sistema de codificación determinado.

report. *informe.* Término genérico aplicado a cualquier análisis de datos impreso producido por un ordenador.

rerun point. *punto de recuperación.* Lugar dentro de una secuencia de instrucciones de un programa que contiene toda la información que se disponga pertinente a la recuperación del mismo. Con ello es posible reconstruir un proceso, si ocurre un error o si es necesario interrumpirlo antes de que éste termine.

reset. *restaurar, poner a cero.* En programación, acción de poner a cero un contador o devolver un indicador a alguna condición estable.

reset pulse. *impulso de puesta en estado inicial.* Impulso cuya finalidad es hacer que un elemento de almacenamiento binario cambie el estado existente en un momento determinado por otro estado fijo.

resident routine. *rutina residente.* Rutina que se encuentra registrada permanentemente en memoria.

residual error. *error residual.* Error generado durante un experimento; es la diferencia entre un resultado exacto calculado teóricamente y uno obtenido empíricamente.

resolution. *resolución.* Término que se aplica al cambio más pequeño susceptible de ser distinguido por un convertidor A/D o ser producido por un dispositivo D/A. Este parámetro puede expresarse en tanto por ciento de la escala completa; si bien por lo general se expresa como el número de bits n en el que el convertidor tiene 2^n estados posibles.

response duration. *duración de respuesta.* Intervalo que abarca el origen de tiempo de un impulso y el momento en que el impulso cae por debajo de un valor operativo específico.

response time. *tiempo de respuesta.* Intervalo de tiempo comprendido entre el momento en que se somete un trabajo a un sistema de cálculo y el instante en que se obtienen los resultados.

restart. *reanudar.* Acción de retornar a un punto previo de un programa para comenzar de nuevo a continuación de un error o de un fallo de máquina.

restore. *restaurar.* Colocar o fijar en un contador, un registro, un conmutador o un indicador a un valor o condición previos. Devolver una dirección variable u otra palabra de ordenador a su valor inicial o seleccionado.

retrieval. *recuperación.* Localización y aislamiento del material almacenado o información específica.

rewind. *rebobinar.* Acción de colocar de nuevo una cinta magnética en el punto de carga.

rewrite. *regrabación, reescritura.* Retención de datos en una zona de almacenamiento. Registrándolos de nuevo en la posición de que se trate después de leerlos en esa posición.

rise time. *tiempo de subida.* Tiempo necesario para que un impulso eléctrico se eleve desde una a nueve décimas de su valor final.

rollback. *repetir.* Ejecutar de nuevo un programa (o parte) en un ordenador.

roll-in. *reincorporar a memoria.* Acción de restaurar en memoria principal programas y datos previamente transferidos desde esta memoria a un almacenamiento auxiliar.

roll-out. *descargar a la memoria externa.* Volcar al exterior del ordenador el contenido de la memoria principal para registrarlo en un sistema de almacenamiento auxiliar.

round. *redondear.* Alterar el valor de los dígitos en el extremo menos significativo de un número a fin de permitir la eliminación de los dígitos al truncar el número.

rounding-off. *redondeo.* Técnica utilizada para representar números y reducir posibles desviaciones a base de incrementar el tamaño de la cantidad.

routine. *rutina.* Parte de un programa, generalmente diseñado para una aplicación específica, cuyo empleo suele ser muy frecuente.

routing. *encaminamiento, encauzamiento.* Asignación de una vía de comunicaciones por la que un mensaje o llamada telefónica alcanzará su destino.

row. *fila.* Cada una de las líneas perpendiculares al eje longitudinal de una cinta de papel o magnética, sobre la que se puede configurar las perforaciones o magnetizaciones de puntos correspondientes a la representación codificada de un carácter. Disposición horizontal de caracteres u otras expresiones.

row pitch. *paso entre filas, paso.* Distancia entre orificios perforados en sentido longitudinal a lo largo de una cinta de papel perforada, medida entre los centros de dos posiciones consecutivas.

run. *pasada de máquina, ejecución de un proceso.* Utilización de un ordenador para la obtención de un trabajo determinado. Pasada de un grupo de tarjetas perforadas por una máquina.
Tratamiento de varias rutinas encadenadas automáticamente, durante el cual no es necesaria la interrupción del mismo por parte del operador.

run phase. *fase de ejecución.* Término relativo a la compilación de programas. Se emplea para indicar el período en que el programa objeto compilado se prueba y procesa por primera vez.

run time. *tiempo de ejecución, tiempo de proceso.* Tiempo durante el cual se procesa un programa. Contrasta con "tiempo de compilación".



S

sample and hold. *muestreo y retención.* Función realizada por un circuito analógico que captura y retiene una señal para luego ser convertida por un convertidor analógico-digital.

sampler. *muestreador.* Conmutador electrónico que se conecta y desconecta a elevada velocidad para producir un tren de impulsos analógicos de muestra.

sampling. *muestreo.* Proceso consistente en registrar el valor de una variable a intervalos de tiempo.

scale. *escala.* Gama de valores aceptados determinada, frecuentemente, por la longitud de palabra del ordenador o por la rutina de que se trate.
Cambio de escala. Proceso consistente en alterar las unidades en que se expresan las variables, con objeto de situarlas al alcance de la capacidad de la máquina o al programa de que se trate.

scan. *explorar.* Examinar cada uno de los elementos de una lista o los registros de un fichero, generalmente como parte de un sistema de recuperación de información en el que se comprueba cada unidad de información en orden a saber si satisface o

no, determinadas condiciones. Acción de verificar el estado o condición de los enlaces de comunicación, o los canales de E/S a fin de determinar si los canales se están utilizando o no.

scanning rate. *frecuencia de exploración.* Frecuencia de muestreo de un ordenador.

scatter-read. *lectura dispersa.* Proceso consistente en distribuir en diversas áreas de la memoria, los datos procedentes de un solo registro de entrada o bloque.

scientific language. *lenguaje científico.* Lenguaje proyectado para escribir programas científicos o matemáticos.

scratch pad memory. *memoria de trabajo.* Area de la memoria reservada para los trabajos intermedios.

scrath tape. *cinta reutilizable.* Cinta que contiene información que ha perdido vigencia y puede borrarse y utilizarse de nuevo para almacenar nuevos datos.

scheduled maintenance. *mantenimiento previsto o programado.* Mantenimiento, generalmente preventivo, que con carácter más o menos periódico se efectúa, según un programa previamente establecido.

scheduler. *planificador.* Programa encargado de supervisar y planificar las tareas a ejecutar por un ordenador multiprogramado.

search. *búsqueda.* Investigación que se hace sobre un conjunto de elementos de información para seleccionar aquel o aquellos que reúnen las condiciones exigidas al ordenar la búsqueda. El elemento investigado puede estar almacenado en cualquier tipo de memoria.

search time. *tiempo de búsqueda.* Tiempo empleado para localizar un registro determinado en un sistema de almacenamiento de datos.

seek. *posicionar.* Búsqueda física en una memoria de acceso aleatorio, situando en posición el mecanismo de acceso con anterioridad a la búsqueda lógica.

segment. *segmentar.* Concepto que expresa la idea de dividir un programa en partes más o menos modulares, para almacenarlo en memoria por partes, sin que para ejecutar el programa sea necesario mantenerlo completo durante el período de tiempo que dure el proceso.

select. *seleccionar.* Elegir según un criterio, uno o más procesos de trabajo o unos dispositivos que se adoptan a un método de trabajo.

selective dump. *vuelco selectivo.* Lectura de un área limitada de memoria.

selector. *selector.* Dispositivo que efectúa una prueba para determinar la presencia de condiciones específicas e inicia las operaciones apropiadas, de acuerdo con el resultado de la prueba.

self checking code. *código de autoverificación.* Sinónimo de código de detección de errores.

self-ressetting loop. *bucle restaurador.* Conjunto de instrucciones secuenciales formando bucle, de las que algunas de ellas tienen por misión sustituir los datos e instrucciones modificadas a su valor inicial, cada vez que se empiece el bucle.

self-triggering program. *programa autoinicializado.* Programa diseñado de forma que su operación comienza automáticamente cuando se carga en el ordenador.

semantics. *semántica.* Ciencia que estudia el significado asignado a la construcción de un lenguaje.

semaphore. *semáforo.* Variable binaria en el sistema operativo que garantiza el acceso a datos no compartibles o a código no reentrante.

sense. *leer, captar, detectar.* Acción de leer las perforaciones existentes sobre una tarjeta perforada. Acción de captar un impulso eléctrico que circula por un circuito. Detectar una condición que reúna unas características determinadas.

sentinel. *centinela, señalizador.* Carácter empleado para indicar la presencia de una condición específica; por ejemplo el final físico de una cinta magnética o el final de un registro de longitud variable en el almacenamiento.

sequence. *secuencia.* Conjunto de términos o instrucciones que se han colocado o dispuesto en un orden definido.

sequential access storage. *almacenamiento de acceso secuencial.* Sistema de almacenamiento en el que el acceso a los datos sólo puede efectuarse en la secuencia en que están almacenados.

sequential control. *control secuencial.* Método de funcionamiento de un ordenador en que las instrucciones se almacenan en el mismo orden en que se ejecutan.

sequential processing. *proceso secuencial.* Proceso de los registros de un fichero de datos que se efectúa de acuerdo con una secuencia predeterminada de claves.

sequential-stacked job control. *control de trabajos agrupados en secuencia.* Sistema de control que asegura que los trabajos se ejecutan en la secuencia en que se presentan al sistema.

serial. *en serie.* Concepto que se aplica a operaciones aritméticas, transferencias, transmisión de datos, etc., y que indica que éstas se ejecutan dígito a dígito o carácter a carácter.

serial D/A converter. *convertidor D/A serie.* Convertidor digital-analógico en el que los datos de entrada digitales se reciben en forma secuencial antes de producir una salida analógica.

serial I/O. *entrada/salida serie.* Tipo de interconexión en el que la información se transmite bit a bit. Las E/S serie más comunes son la RS-232 y el bucle de corriente de 20 mA.

serial number. *número de serie.* Número que se adjunta a una información para reconocer la posición que dicha información ocupa en una serie ordenada de ellas.

serial transfer. *transferencia en serie.* Transferencia sucesiva de una serie de elementos de información, de unidad a unidad o de terminal a terminal.

serviceability. *indice de utilidad.* Fiabilidad del equipo, basada en algún criterio objetivo. Se adoptan diferentes criterios para determinar o valorar el índice de utilidad.

set. *conjunto, juego.* Colección o conjunto de elementos que tienen alguna característica en común o entre las cuales existe alguna relación.

settling time. *tiempo de establecimiento.* Tiempo transcurrido desde la aplicación de una entrada escalonada a fondo de escala a un circuito hasta el tiempo en que la salida ha entrado y permanecido dentro de una banda de error especificada en torno a su valor final.

setup. *preparación, puesta a punto.* Serie de operaciones con las que se prepara a las unidades que componen un equipo para su funcionamiento.

shift. *desplazar.* Operación consistente en trasladar o mover los elementos de una unidad de información hacia la izquierda o hacia la derecha.

shift out. *desplazar de salida.* Acción de mover la información, dentro de un registro general, hacia uno de los extremos, a fin de que, a medida que la información sale por ese mismo extremo, los ceros vayan entrando por el extremo opuesto.

shift register. *registro de desplazamiento.* Registro en que los datos almacenados se pueden someter a un desplazamiento hacia la derecha o hacia la izquierda.

significant digits. *dígitos significativos.* Dígitos o posiciones de dígitos de un número cuyos valores se conocen y tienen relación con la precisión del número.

simplex. *simplex, unidireccional.* Canal de comunicaciones que permite la transmisión en un sentido solamente.

simulator. *simulador.* Sistema diseñado para efectuar la simulación de un proceso en tiempo real.

single-ended amplifier. *amplificador de un solo extremo.* Amplificador que desarrolla solamente una señal de salida.

skip. *saltar.* Omitir, saltar o eludir una o más instrucciones en una secuencia de instrucciones.

slew rate. *pendiente.* Velocidad máxima de variación en la salida de un amplificador operacional. La pendiente está limitada por corrientes de carga internas y capacidades; por lo general se expresa en voltios por nanosegundo.

slice. *limitar.* Acción de eliminar las partes de una forma de onda que quedan fuera de unos límites de amplitud dados en el mismo lado del eje cero.

slicer. *circuito amplificador de impulsos, limitador.* Circuito que amplifica de forma eficaz una parte de los impulsos de llegada comprendidos entre dos niveles de amplitud espaciados entre sí con mucha proximidad.

smooth. *ajustar, nivelar, filtrar.* Acción de aplicar procedimientos en orden a disminuir o eliminar las fluctuaciones rápidas en los datos.

snapshot dump. *vuelco instantáneo.* Vuelco de partes seleccionadas del almacenamiento que se puede producir en diversos puntos durante el desarrollo de un programa, generalmente a efectos de depuración.

software. *soporte lógico, conjunto de programas, software.* Conjunto de programas y procedimientos que se incluyen en un equipo de tratamiento de datos y que

hace posible la utilización eficaz del mismo.

sort. *clasificar.* Acción de disponer las unidades o elementos de información en grupos, según las claves de identificación de cada uno. Los elementos se ordenan en secuencia si la disposición de las claves sigue algún orden predeterminado.

source code. *código fuente.* Programa escrito del usuario, usualmente en código ASCII, y que se introduce en el sistema.

source language. *lenguaje fuente.* Lenguaje original, utilizado por el programador, sobre el cual opera un programa traductor.

stack. *pila.* Estructura LIFO que memoriza la información en orden cronológico.

stand-alone. *autónomo, independiente.* Dispositivo o sistema que opera con independencia de otro dispositivo, sin su ayuda o sin estar conectado a él.

standby. *en espera, espera.* Condición del equipo que permite reanudar por completo el funcionamiento uniforme y regular en un breve plazo de tiempo.

standby computer. *ordenador de reserva.* Ordenador utilizado en un sistema dual o dúplex que está a la espera para hacerse cargo de la carga de proceso en tiempo real cuando sea necesario.

statement. *sentencia.* Instrucción en lenguaje fuente. También se emplea en cualquier expresión que se pueda introducir en un compilador, incluyendo las sentencias de comentarios y las pseudoinstrucciones que controlan el funcionamiento de la compilación. Una vez compiladas, las sentencias generalmente dan por resultado varias instrucciones en código de máquina.

state table. *tabla de estados, tabla de transición de estado.* Lista de salidas de un circuito lógico basado en las entradas y en previas salidas. Dicho circuito tiene memoria y no puede ser descrito por una simple tabla de verdad.

static dump. *vuelco estático.* Vuelco que se realiza cuando un programa llega a un fin de pasada o a alguna otra fase reconocible dentro del proceso.

static memory. *memoria estática.* Memoria MOS que utiliza un multivibrador biestable como elemento de almacenamiento. No necesita regrabado y no requiere reloj; además tampoco pierde su contenido siempre y cuando no se corte la alimentación.

status. *estado.* Condición actual de un dispositivo.

status register. *registro de estado.* Registro utilizado para mantener la información de estado dentro de una unidad funcional, como una MPU, una PIC, etc. Un registro de estado de MPU típico proporciona indicación de acarreo, rebasamiento, signo, cero e interrupción. Puede también incluir paridad, desinhibición o máscara.

step. *ejecutar un paso.* Acción de hacer que un ordenador ejecute una operación.

stop bit. *bit de parada.* Bit que indica el final de una transmisión serie asíncrona.

stop signal. *señal de parada.* Señal utilizada para poner un receptor en reposo para prepararlo para recibir la señal siguiente.

storage. *almacenamiento, memoria.* Dispositivo diseñado para aceptar la introducción y retención de datos para su posterior recuperación.

storage cell. *celdilla de almacenamiento.* Unidad elemental de almacenamiento.

stored program. *programa almacenado.* Programa totalmente contenido en memoria y que se puede alterar dentro de ella.

straight-line coding. *codificación rectilínea.* Codificación que evita el empleo de bucles, mediante repetición de partes de la codificación, siempre que sea necesario.

stream. *flujo, corriente.* Datos que se están transfiriendo a/desde un medio de almacenamiento externo, representado como una serie de partidas de datos en forma de caracteres y prescindiendo de los límites de la línea de impresión o del tamaño del registro de entrada.

string. *serie, tira.* Cualquier conjunto de elementos o unidades que han sido ordenados en una secuencia, según un orden específico. Cualquier conjunto de caracteres o dígitos consecutivos que se encuentran en el almacenamiento.

string length. *longitud de la serie.* En la clasificación, número de registro de una serie.

strobe. *muestreo.* Señal de selección que se activa cuando los datos en una barra son correctos.

stroke. *traza, segmento.* Punto o marca utilizado en la formación de caracteres. Elemento del trazado de un carácter impreso.

stunt box. *supresor de impresión, caja reguladora.* Parte de un teleimpresor que codifica las señales empleadas para controlar el funcionamiento de la máquina, en contraste con la información que se ha de imprimir.

style. *estilo.* En el reconocimiento de caracteres ópticos, dícese de las proporciones que distinguen a los caracteres y que permanecen constantes, cualquiera que sea el tamaño del carácter.

subroutine. *subrutina.* Parte de un programa que ejecuta una parte o sección lógica de las funciones generales del programa y que está disponible siempre que se necesite ese juego específico de instrucciones.

subset. *subconjunto.* Conjunto o grupo de elementos que guardan determinadas relaciones entre sí y forman parte de un conjunto más importante o de jerarquía superior. Sinónimo de modem.

substraction. *sustracción.* Operación aritmética en la que un operando (sustraendo) se resta de otro (minuendo) para formar la diferencia.

successive approximation register. *registro de aproximaciones sucesivas.* Circuito de control digital usado para controlar el funcionamiento de un convertidor A/D de aproximaciones sucesivas.

sum. *suma.* Resultado obtenido al añadir a un número (addend: primer sumando) otro número (augend: incremento).

summer. *sumador. sumador analógico.* Dispositivo con dos o más entradas analógicas variables y otra de salida, que es la suma de las de entrada.

support programs. *programas de apoyo.* Conjunto de programas destinados a apoyar los típicos de un sistema operativo.

symbol. *símbolo.* Cualquier carácter, conjunto de caracteres o cifra, aceptados convencional o arbitrariamente como representativos de alguna cantidad, instrucción, etc.

symbolic coding. *codificación simbólica.* Escritura de un programa en un lenguaje fuente.

synchronization character. *carácter de sincronización.* Carácter que se inserta automáticamente en la corriente de datos de un equipo de comunicaciones síncrono para mantener y establecer el sincronismo.

synchronizer. *sincronizador.* Dispositivo de almacenamiento que actúa como memoria intermedia para contrarrestar los efectos de la transmisión de datos entre dispositivos que operan a velocidades distintas.

syntax. *sintaxis.* Conjunto de reglas que rigen la escritura de las sentencias o expresiones del lenguaje fuente.



table. *tabla.* Conjunto de datos en que cada unidad elemental de los mismos se identifica por una clave, los datos están dispuestos en forma apropiada para una fácil consulta.

table look-at. *investigación de tablas.* Localización de un elemento de una tabla por cálculo directo en lugar de efectuar una investigación como en la consulta de tablas.

table look-up. *consulta de tablas, búsqueda de tablas.* Método de investigación de tablas para localizar elementos relacionados con una clave determinada. Operación de la búsqueda de la función de una tabla en memoria que corresponde a un argumento determinado.

tabulate. *tabular.* Acción de ordenar datos disponiéndolos en forma de tabla. Acción desarrollada para imprimir totales.

tag. *rótulo, distintivo.* Símbolo anexo a un elemento, utilizado para su identificación.

tag format. *formato de la etiqueta.* Disposición y escritura de un registro empleado como etiqueta para localizar una posición de excedentes.

takedown. *desmontaje.* Operaciones efectuadas al término de un ciclo de funcionamiento para dejar preparado el equipo para la siguiente puesta a punto.

tally. *cuenta, recuento.* Lista impresa de cifras que produce una máquina sumadora.

tape. *cinta.* Término genérico por el que se conoce a los soportes de información formados por una cinta magnética o de papel.

tape drive. *impulsor de cinta.* Dispositivo perteneciente a una unidad de cintas magnéticas, cuya finalidad es mover las

cintas bajo las cabezas de lectura-escritura.

tape file. *fichero de cinta.* Conjunto secuencial integrado por registros grabados en cinta.

tape reader. *lector de cinta.* Dispositivo electromecánico cuya finalidad es detectar las perforaciones representativas de la información codificada existente en una cinta de papel. Dispositivo capacitado para detectar la información registrada en una cinta magnética.

tape unit. *unidad de cinta.* Unidad formada por un transportador de cintas, una cabeza lectora-escritora y los dispositivos correspondientes de control, cuyo conjunto permite tratar las cintas magnéticas en un ordenador.

tape wound core. *núcleo de bobinado magnético.* Núcleo magnético constituido por un bobinado de cinta ferromagnético.

target language. *Lenguaje resultante. lenguaje objeto.* Lenguaje al cual se traduce una sentencia o programa.

target phase. *fase objeto.* Durante la compilación, fase en la que se procesa primeramente el programa objeto.

target program. *programa resultante.* Sinónimo de programa objeto.

task. *tarea.* Unidad básica de multiprogramación bajo el programa de control. Ejecución de uno o más procedimientos por parte de un solo flujo de control.

telematique. *telemática.* Término originado para describir la combinación de ordenadores con las telecomunicaciones.

teleprocessing. *teleproceso.* Término registrado por IBM que se emplea para describir sistemas en los que se conectan localidades distantes a un ordenador central por medio de circuitos de transmisión de datos.

telesoftware. *telesoftware, software a distancia.* Transmisión o radiodifusión de software para su recepción por terminales de videotex inteligentes.

Teletel. *Teletel.* Nombre dado en Francia al servicio público de videotex.

teletype. *teletipo (marca registrada).* Dispositivo diseñado para transmitir mensajes a distancia y recibirlos en forma impresa. La transmisión se efectúa mediante teclado o por cinta de papel perforada.

teletypewriter. *teleimpresora.* Dispositivo terminal telegráfico, parecido a una máquina de escribir, que se emplea para transmitir y recibir mensajes en un sistema de comunicaciones telegráficas.

telex. *telex.* Servicio automático de intercomunicación que se emplea para la comunicación entre abonados utilizando un equipo telegráfico, tal como los teleimpresores.

Telidon. *Telidon.* Sistema de videotex del Departamento Canandiense de Comunicaciones.

temporary storage. *almacenamiento temporal.* Posiciones de memoria reservadas para los resultados intermedios.

tens complement. *complemento a diez.* Operación resultante de sustraer cada dígito de un número a la base del sistema menos uno, añadiendo después 1 al último dígito significativo.

terminal. *terminal.* Dícese de cualquier punto en el que se pueden introducir o extraer datos de un sistema de comunicación de datos. También se conoce como terminal de datos.

terminal repeater. *repetidor terminal.* Repetidor que se utiliza en el extremo de una línea de enlace.

terminate. *terminar, concluir, finalizar, suspender, interrumpir.*

test. *probar, examinar.* Acción de examinar un elemento de datos o un indicador para determinar si satisface alguna condición predeterminada.

test program. *programa de verificación.* Programa especialmente diseñado para demostrar que un equipo está en condiciones de ejecutar un proceso específico.

test routine. *rutina de verificación.* Rutina especialmente concebida para demostrar que un equipo está en condiciones de efectuar un determinado proceso.

test run. *pasada de prueba.* Prueba que se realiza para comprobar que un programa determinado está funcionando correctamente, y en la que se emplean datos de prueba para generar resultados que se comparan con las respuestas que se espera obtener.

text. *texto.* Elemento de información de cualquier mensaje, excluyendo aquellos caracteres o bits necesarios para facilitar su transmisión.

text buffer. *memoria intermedia de textos.* Memoria tampón que contiene única y exclusivamente el texto de los mensajes.

three-input adder. *sumador con tres entradas.* Unidad sumadora capaz de recibir tres señales de entrada (una que representa el primer sumando, otra el segundo y la tercera el dígito de arrastre) y de emitir dos señales de salida (una representa el dígito suma y otra el dígito que ha de arrastrarse a la operación siguiente).

three-input substracter. *restador con tres entradas.* Unidad restadora capaz de recibir tres señales de entrada (una que representa el minuendo, otra el sustraendo y la tercera el dígito de arrastre) y de emitir dos señales de salida (una que representa al dígito diferencia y otra al de arrastre, con el que ha de operarse en la siguiente posición de dígito).

three-plus-one address. *dirección de tres más uno.* Relativo a una instrucción que contiene tres direcciones de operando y una dirección de control.

threshold. *umbral.* Valor específico establecido para controlar la entrada desde un elemento umbral.

threshold element. *elemento umbral.* Dispositivo que recibe una serie de señales digitales de valor 1 o 0 afectadas cada una de ellas de un peso específico diferente, de las que obtiene otra señal binaria de salida, cuyo estado depende de la suma de los pesos específicos indicados y del valor 1 o 0 de dichas señales de entrada.

throughput. *rendimiento específico, productividad.* Productividad de una máquina, sistema o procedimiento, medidos según un factor de comparación que tenga significado para el proceso de que se trate.

time access. *tiempo de acceso.* Tiempo invertido en la búsqueda de una información situada en memoria y su transferencia a la ALU del ordenador.

time-division multiplexing. *Multiplexado por división de tiempo.* Sistema por el que un canal se pone a disposición de un número determinado de dispositivos terminales, cada uno de los cuales ocupa el canal, para la transmisión de datos, durante períodos cortos y a intervalos regulares.

time-share. *tiempo compartido.* En términos generales, dícese de la utilización de un dispositivo, equipo u ordenador, por dos o más aplicaciones entremezcladas entre sí.

time-slicing. *División o reparto del tiempo.* Sinónimo de tiempo compartido.

timing. *sincronización, temporización.* Proceso de cálculo y regularización de unas operaciones con respecto a un ritmo de tiempo.

timing diagram. *cronograma.* Diagrama en el que se reflejan las variaciones de una o más señales en función del tiempo, relacionándolas entre sí.

trace. *rastreo, diagnóstico, trazado.* Análisis de los resultados obtenidos tras la ejecución de cada instrucción, con el fin de obtener un diagnóstico de funcionamiento directo.

track. *pista.* En un dispositivo de memoria magnética, canal que sirve para registrar datos.
Uno de los canales para registrar datos en forma de perforación en una cinta de papel.

track-and-hold (T/H). *seguimiento y retención.* Circuito de muestreo y retención que puede continuamente seguir la señal de entrada en el modo de muestreo y posteriormente, bajo orden, pasar a modo retención.

tracking A/D converter. *convertidor A/D de seguimiento.* Convertidor de tipo contador que puede seguir de forma continuada la entrada analógica a cierta velocidad máxima especificada y actualizar en continuo su salida digital a medida de que varía la señal de entrada.

track pitch. *paso entre pistas, separación entre pistas.* Distancia que separa dos pistas contiguas.

transfer. *transferir.* Acción de transmitir uno o varios datos de un punto a otro, escribiéndolos con idéntico contenido en el receptor. Mover información desde unas posiciones de memoria a otras.

transfer function. *función de transferencia.* Expresión matemática que especifica la relación entre dos fenómenos que existen en puntos diferentes, en tiempo o en espacio, en un sistema determinado.

transfer instruction. *instrucción de transferencia.* Instrucción que copia datos de una parte de memoria a otra. Instrucción de bifurcación que sirve para transferir el control de una parte de un programa a otro.

transfer time. *tiempo de transferencia.* Intervalo de tiempo comprendido entre el instante en que se inicia la transferencia de los datos a/o desde un dispositivo de almacenamiento y el momento en que se completa dicha transferencia.

transient. *transitorio.* Perturbación física intermedia que se produce entre dos condiciones correspondientes a estados estables o permanentes.

transition. *transición.* Cambio, en un circuito, de una condición operativa a otra.

translator. *traductor.* Programa que convierte las sentencias escritas en un lenguaje de programación al formato de otro lenguaje de programación. Por ejemplo, de un lenguaje fuente a código máquina.

transmission. *transmisión.* Transferencia eléctrica de una información desde un punto a otro.

transmission loss. *pérdida de transmisión.* Sinónimo de atenuación.

transmission speed. *velocidad de transmisión.* Número de elementos de información enviados por unidad de tiempo.

transparent. *transparente.* Proceso que no es visible al usuario o a otros dispositivos.

trap. *desvío.* Operación de bifurcación que el equipo físico inicia automáticamente al detectar alguna condición anormal durante el proceso de un programa.

tree structure. *estructura en árbol.* Disposición de una base de datos en un número determinado de niveles jerárquicos, de los que los de nivel inferior presentan la información en menor detalle.

tri-state. Marca registrada por National Semiconductor, consistente en una configuración de salida de algunas familias lógicas, capaz de asumir tres estados: lógico alto, lógico bajo y alta impedancia.

trouble-shooting. *localización de errores, investigación de averías.* Investigación o búsqueda de los errores de un programa o de la causa que ha producido algún fallo de máquina.

truncate. *truncar.* Acción consistente en suprimir los dígitos de un número que no son significativos, de acuerdo con algún requisito predeterminado en cuanto a la exactitud del resultado.

truth table. *tabla de decisión lógica, tabla de verdad.* Tabla que describe una función lógica mediante el listado de todas las combinaciones lógicas posibles de valores de entrada y la indicación de los valores verdaderos de salida que corresponden a cada combinación.

turnaround time. *tiempo de respuesta.* Tiempo que se necesita para completar una tarea; por ejemplo, para recoger los datos, transcribirlos para su proceso, realizar el cálculo y proporcionar el resultado al usuario.

two-address instruction. *Instrucción de dos direcciones.* Formato de una instrucción en cuyo contenido se incluyen dos direccionamientos.

two-core-per-bit store. *memoria de dos núcleos por bit.* Concepto que describe una memoria en la que cada dígito binario se representa por dos núcleos magnéticos.

two-input adder. *sumador de dos entradas.* Unidad de un componente aritmético con posibilidad de recibir dos señales de entrada que representan un dígito de un número y otro que corresponde al primer sumando (o al dígito de arrastre) y dos señales de salida que representan el dígito suma correspondiente y el de arrastre que debe operar en la siguiente operación.

two-input substracter. *restador con dos entradas.* Unidad de un componente aritmético con posibilidades de recibir dos señales de entrada que representan un dígito de un número y otro que corresponde al sustraendo (o al dígito de arrastre) y dos señales de salida que representan al dígito diferencia correspondiente y al de arrastre que debe operar con la siguiente posición de número.

two-level subroutine. *subrutina de dos niveles.* Subrutina que contiene otra subrutina dentro de su propia escritura.

two-out-of-five code. *Código de dos a cinco.* Código binario en que cada dígito decimal se representa por dos bits 1 y tres bits 0.

two's complement. *complemento a dos.* Método de expresar números binarios en donde el negativo de un número se genera complementando al número y añadiendo 1.

type bar. *barra de tipos.* Elemento de tipo lineal que contiene todos los símbolos imprimibles.

type wheel. *rueda de tipos.* Componente de una impresora de líneas en que los caracteres están contenidos en una rueda que se posiciona por giros en sentido vertical.

U

UART. *transmisor receptor asíncrono universal. UART.* Convertidor serie-paralelo y paralelo-serie.

unary operation. *operación unitaria.* Operación que se ejecuta sobre un solo operando.

unattended operation. *funcionamiento sin operador.* Tipo de operación que, por los dispositivos automáticos de que está dotada una estación terminal, permite realizar la transmisión y recepción de datos sin la intervención humana.

unattended station. *estación sin operador.* Estación repetidora que normalmente no está dotada de personal encargado de su atención y mantenimiento.

unconditional jump. *salto incondicional.* Salto que se produce en la ejecución de la secuencia normal de unas instrucciones al aparecer una instrucción de transferencia incondicional.

unitary code. *código unitario.* Código que consta de un sólo código. La cantidad que representa está determinada por el número de veces que el código se repite.

unpack. *desempaquetar, desagrupar.* Recuperar datos originales de una posición de almacenamiento en que han sido condensados en unión de otros datos.

unused time. *tiempo de no utilización.* Tiempo durante el cual el equipo está desconectado y fuera de servicio.

update. *actualizar.* Proceso de modificación de un fichero o archivo con otra información utilizada en un tratamiento.

uptime. *tiempo productivo, tiempo activo.* Período de tiempo durante el cual un equipo está trabajando o está disponible para efectuarlo.

USRT. *transmisor receptor síncrono universal.* Convertidor serie-paralelo para comunicaciones de alta velocidad.

utility program. *programa de utilidad.* Programa cuya estructura está dirigida a prestar un servicio en la ejecución de otro programa, generalmente de aplicación. Por ejemplo, un programa de entrada.

utility routine. *rutina de utilidad.* Tipo de rutina utilizada para colaborar en el proceso de un ordenador y para tratar las operaciones de máquina necesarias para el proceso de datos, pero que no contribuye directamente a la producción de resultados.



validity check. *verificación de validez.* Verificación o reconocimiento realizado para contrastar la presencia o ausencia de error.

variable. *variable.* Cantidad que puede adquirir un valor cualquiera dentro de una serie de valores.

variable block. *bloque variable.* Bloque cuyo tamaño no es fijo sino que varía (dentro de ciertos límites) de acuerdo con las necesidades de los datos.

variable-length instruction. *instrucción de longitud variable.* Característica que da por resultado un mayor aprovechamiento de la memoria principal, al utilizarse únicamente las posiciones de memoria exigidas por las instrucciones de la aplicación de que se trate.

verify. *verificar.* Acción de comprobar una transcripción realizada con unos datos. Acción de comprobar, mediante una repetición de la misma operación, si la perforación realizada sobre unas tarjetas o la grabación hecha sobre unas cintas magnéticas por una grabadora, son correctas o no.

video A/D converter. *convertidor A/D de video.* Convertidor ultrarrápido que permite obtener velocidades de conversión de 5 MHz e incluso mayores.

Viewdata. *Viewdata.* Término genérico utilizado en Gran Bretaña para denotar sistemas videotex.

virtual address. *dirección virtual.* En los sistemas dotados de memoria virtual, dícese de la dirección que hace referencia a la memoria virtual y ha de convertirse a dirección de la memoria real como condición previa a su utilización.

volatile storage. *memoria volátil.* Sistema de memoria en que los datos almacenados se pierden cuando se desconecta la corriente que alimenta al sistema.

voltage-to-frequency converter. *convertidor tensión-frecuencia.* Dispositivo que convierte una tensión analógica en un tren de impulsos digitales con una frecuencia proporcional a la tensión de entrada.



waiting time. *tiempo de espera.* Tiempo que transcurre entre el momento en que la unidad de control ordena una transferencia de datos hacia o desde la memoria del sistema, y el preciso instante en que ésta comienza.

walk down. *pérdidas acumuladas.* En una memoria que funciona incorrectamente, los impulsos excitadores parciales o los impulsos de dígito sucesivos provocan un proceso magnético irreversible en una celda magnética. Este proceso se conoce como pérdidas acumuladas o pérdidas de información.

waste instruction. *instruccion no operativa.* Instrucción que se incorpora a un programa para asegurar la secuencia de otras instrucciones de dicho programa, preparando cambios futuros en el mismo o cumplimentando ciertas condiciones necesarias en éste (completar un bloque de instrucciones, por ejemplo), pero sin especificar operación alguna.

Whetstone. *whetstone.* Acotación de un solo programa utilizado para medir las prestaciones en coma flotante de un procesador.

willful intercept. *interceptación premeditada.* Acción por la que se interceptan mensajes destinados a estaciones terminales que están experimentando dificultades o anomalías de funcionamiento en el equipo o en la línea.

word. *palabra.* Unidad lógica de información que puede tener cualquier número de bits, pero normalmente es de 4, 8 o 16.

word length. *longitud de palabra.* Número de bits o caracteres que integran una palabra de máquina.

word mark. *marca de palabra.* Símbolo utilizado para indicar el principio o fin de una palabra de máquina.

word time. *tiempo de palabra.* Tiempo necesario para transferir una palabra desde una posición de memoria a otra o desde un dispositivo de almacenamiento a otro.

work area. *área de trabajo.* Zona de memoria cuyo cometido es el almacenamiento temporal de los datos durante el curso de un proceso.

write. *escribir, grabar.* Acción de transcribir datos en una forma de almacenamiento desde otra de almacenamiento distinta; por ejemplo, transcribir datos a una cinta magnética desde la memoria principal de un ordenador.

write pulse. *impulso de escritura.* Impulso excitador que establece la condición 1 de una celda magnética o graba en ella.

write time. *tiempo de escritura.* Intervalo de tiempo que transcurre entre el instante en que comienza la transcripción a un dispositivo de almacenamiento y el instante en que se termina.

write up. *documentación de programa.* Conjunto de documentos necesarios para la utilización de un programa.



X-off. *transmisor desconectado.*

X-on. *transmisor conectado.*

X punch. *perforación X.* Orificio que se perfora en la posición X de una ficha perforada (por lo general, la segunda fila empezando por la parte superior).

xerographic printer. *impresora xerográfica.* Dispositivo de impresión que utiliza xerográficas y en el que se determina el formato de la impresión a realizar antes de ejecutarla.

X-Y plotter. *trazador de gráficos X-Y.* Dispositivo que se utiliza en combinación con un ordenador para trazar puntos de coordenadas en forma de gráfico.

Xtal. *cristal. Véase cristal.*



Y-punch. *perforación Y.* Orificio que se perfora en la posición Y de una ficha (por lo general, en la fila superior). Posición Y.



zero. *cero, nada.* Número que denota magnitud cero. Condición de códigos que un ordenador reconoce como cero.

zero access storage. *memoria de tiempo de acceso cero.* Memoria cuyo tiempo de latencia o de espera es muy breve. Este término, utilizado frecuentemente en otro tiempo, está perdiendo aceptación en la actualidad, ya que implica un concepto falso o erróneo.

zero address instruction format. *formato de instrucción sin dirección.* Instrucción en cuya constitución no existe la parte de direccionamiento.

zero fill. *rellenar con ceros.* Rellenar con caracteres utilizando la representación de cero.

zeroise. *poner a ceros.* Restaurar un registro mecánico o electrónico restituyéndolo a su posición o estado cero.

zero flag. *señalizador de cero.* Señalización del estado de la ALU que indica que la operación anterior era cero.

zero output signal. *señal de salida cero.* Salida que proporciona una celda magnética en el estado o condición cero cuando se le aplica un pulso de lectura.

zone. *zona.* Area que se reserva en memoria para efectuar un trabajo específico.

zone bits. *bits de zona.* Bits que representan la parte no numérica de un carácter. Bits distintos de los cuatro que se emplean para representar un dígito en un código de octeto.

zone punch. *perforación de zona.* Perforación realizada en las filas 12, 11 y 0 de una ficha perforada. Perforación distinta a la numérica.

los autores

José Bosch Solsona. Ingeniero de Telecomunicación por la Escuela Técnica Superior de Ingenieros de Telecomunicación de Barcelona en 1977. Desde este año es profesor del departamento de Ordenadores en la E.T.S.I. de Telecomunicación de Barcelona.

Sus trabajos se sitúan dentro del campo de los microcomputadores y más en especial en las pequeñas redes de microcomputadores, realizando actualmene la tesis doctoral en este tema.

Josep M.ª Fuertes i Armengol. Doctor Ingeniero Industrial por la ETSEI de Barcelona. Ha colaborado con el Institut de Cibernética (UPC-CSIC) desde 1974 a 1986, donde ha sido responsable del área de electrónica. Ha sido profesor de la ETSEIB desde 1975 a 1986, y actualmente es Profesor Titular del Departamento de Automática y Sistemas Híbridos de la Facultad de Informática de Barcelona.

Ha presentado diversos trabajos en Congresos Internacionales sobre el tema de microprocesadores y sistemas distribuidos de control y es coautor del libro Electrónica y Automática Industriales de la serie Mundo Electrónico.

Miguel García Hoffmann. Obtuvo el grado de Doctor Ingeniero Industrial con premio extraordinario en la E.T.S.I.I.B. Entre 1970-79 desarrolló tareas de investigación aplicada en el Instituto de Cibernética del Consejo Superior de Investigaciones Científicas y la U.P.B., fundamentalmente en el campo de la Electrónica Digital.

Ha sido profesor del Departamento de Arquitectura de Computadores de la Facultad de Informática de Barcelona, y actualmente se encuentra en la cátedra de Métodos Informáticos de la E.T.S.I.I.B. Las áreas fundamentales de su interés son las redes de computadores, protocolos de comunicación y los procesos concurrentes en general. Forma parte del IFIP Working Grupo 6.1 «Packed-Switched Network Interworking», Study Group C, «formal description and verification».

Víctor M. Gil Galés. Ingeniero Industrial. Actualmente desarrolla actividad de jefe de proyecto de Entel, S.A., en aplicaciones de proceso de datos y control de procesos. Ha trabajado durante varios años en el laboratorio de Investigación y Desarrollo de Telesincro, S.A., como responsable de proyecto en diseño de ordenadores de gestión y terminales basados en microcomputadores. Ha desarrollado actividad docente como encargado de curso y de prácticas en las cátedras de Automática y Electrónica de la E.T.S.I.I. de Barcelona, en cuyos laboratorios ha participado en diversos proyectos de investigación.

Enrique Herrada Lillo. Ingeniero de Telecomunicación por la Escuela Técnica Superior de Ingenieros de Telecomunicación de Madrid en 1973. Desde este año es profesor del Departamento de Ordenadores en la E.T.S.I. de Telecomunicación de Barcelona.

Sus trabajos se sitúan dentro del campo de los microcomputadores y concretamente en el estudio de métodos de verificación y aumento de la fiabilidad.

Santiago Insa Alfaro. Cursó estudios superiores en la E.T.S.I.I.B. en la especialidad Eléctrica. Fue colaborador en el Laboratorio de Automática de dicha Escuela Técnica, y profesor de las asignaturas de Servomecanismos y Electrónica. En la actualidad trabaja en el departamento de Investigación de la empresa Telesincro, S.A. como jefe de proyecto. Así mismo colabora en la Facultad de Informática de Barcelona de la U.P.B. en la rama «Hardware».

Manuel Medina Llinás. Ingeniero de Telecomunicación por la Universidad Politécnica de Madrid desde 1974. Ha disfrutado de una beca de formación de personal investigador del M.U.I. durante tres años. Ha realizado estudios sobre el diseño de sistemas basados en microprocesador en la Escuela Politécnica de Lausanne y en el laboratorio de ordenadores de la E.T.S.I. de Telecomunicación de Barcelona, donde actualmente es profesor adjunto.

Joan Navarro Blanco. Natural de Barcelona. Ha cursado estudios en la U.L. Tarragona, E.T.P.I. Madrid y E.T.S.I.I. Barcelona. Desde hace 10 años trabaja en el Dto. Investigación y Desarrollo de Telesincro, S.A., donde ha ocupado diversos cargos como responsable de proyecto, Director del Dto. de Electrónica y Director de Innovación Tecnológica.

José María Pallarés Curto. Ingeniero de Telecomunicación por la Escuela Técnica Superior de Ingenieros de Telecomunicación de Barcelona en 1977. Desde este año es colaborador del Departamento de Ordenadores en la E.T.S.I. de Telecomunicación de Barcelona.

Sus trabajos se sitúan dentro del campo de los microcomputadores y actualmente presta sus servicios como técnico en sistemas microcomputadores en la firma Interface S.A. de Barcelona.

Ramón Pericas Bosch. Ingeniero Industrial (promoción 1977 por Barcelona). Estudió en el Instituto Tecnológico de Georgia (USA) en donde obtuvo el título de Master of Science in Electrical Engineering (1977). Actualmente es profesor de Electrónica General y Electrónica Industrial en la E.T.S.I.I.B. así como del Laboratorio de Investigación y Desarrollo de Electrónica de la misma Escuela, en el campo de aplicaciones industriales de los microprocesadores e instrumentación industrial.

Eugenio Rey Veiga. Oficial Radio de 2.ª por la Escuela Superior de la Marina Civil de Barcelona (1973) y de 1.ª clase —promoción 1975— en la misma. Tras tres años de actividad profesional pasa a ocupar el cargo de redactor en Mundo Electrónico. Autor de numerosos artículos de electrónica profesional, es coautor de cuatro libros de Marcombo y formó parte del equipo redactor de la versión castellana del «Diccionario de Términos Científicos y Técnicos» de McGraw-Hill/Boixareu (1981). Fue responsable de la sección «Micromundo» de Mundo Electrónico desde 1980 hasta 1986, tarea que simultaneó con el cargo de redactor-jefe de dicha publicación. Miembro de AIPET, desde julio de 1986 es director de Productrónica de Boixareu Editores.

Mateo Valero Cortés. Ingeniero Superior de Telecomunicación por la Escuela Técnica Superior de Ingenieros de Telecomunicación de Madrid en 1974. Doctor Ingeniero de Telecomunicación por la Escuela Técnica Superior de Ingenieros de Telecomunicación de Barcelona en 1980. Desde 1974 profesor del Departamento de Ordenadores en la E.T.S.I. de Telecomunicación de Barcelona. Becario del M.U.I. para la realización de tesis doctoral. Becario del Gobierno francés habiendo realizado dos estancias en el Equipo de Arquitectura de Ordenadores del E.N.S.I.M.A.G. en Grenoble.

Sus trabajos se sitúan dentro del campo de los microcomputadores y más en especial en los Sistemas Distribuidos. Realizó la tesis doctoral sobre el tema de Redes Locales de Microcomputadores. Actualmente desarrolla su investigación sobre los Sistemas Multimicroprocesadores dentro del Departamento de Arquitectura de Computadores de la Facultad de Informática de Barcelona.

TITULOS DE LA «SERIE: MUNDO ELECTRONICO»

MICROPROCESADORES Y MICROCOMPUTADORES
344 páginas. 432 figuras. 21,5 × 28,5 cm.

TRANSDUCTORES Y MEDIDORES ELECTRONICOS
288 páginas. 525 figuras. 21,5 × 28,5 cm.

INTERCONEXION DE PERIFERICOS A MICROPROCESADORES
240 páginas. 296 figuras. 21,5 × 28,5 cm.

ELECTRONICA Y AUTOMATICA INDUSTRIALES (2 volúmenes)
668 páginas. 1.298 figuras. 21,5 × 28,5 cm.

ENERGIA SOLAR FOTOVOLTAICA
252 páginas. Ilustrado. 21,5 × 28,5 cm.

MANUAL DE ALTA FIDELIDAD Y SONIDO PROFESIONAL
244 páginas. 384 figuras. 21,5 × 28,5 cm.

MANUAL DEL RADIOAFICIONADO MODERNO
368 páginas. 589 figuras. 21,5 × 28,5 cm.

TELEVISION DIRECTA POR SATELITE
100 páginas. 86 figuras. 21,5 × 28,5 cm.

TELEINFORMATICA Y REDES DE COMPUTADORES
352 páginas. 388 figuras. 21,5 × 28,5 cm.

MICROELECTRONICA
384 páginas. 568 figuras. 21,5 × 28,5 cm.

COMUNICACIONES POR FIBRA OPTICA
180 páginas. 205 figuras. 21,5 × 28,5 cm.

SISTEMAS CAD/CAM/CAE
400 páginas. 504 figuras de las cuales 18 en color. 21,5 × 28,5 cm.

INTELIGENCIA ARTIFICIAL
296 páginas. 214 figuras. 21,5 × 28,5 cm.

OTRAS OBRAS MARCOMBO DE INTERES

SERVICIO DE INFORMACIÓN PERIÓDICA

MARCOMBO, S.A., pensando en usted, ha creado un servicio para que reciba cómodamente, de **forma gratuita**, sin ningún compromiso y en la dirección que nos indique, la más amplia información acerca de los temas de su interés y que edita MARCOMBO, S.A.

Para beneficiarse de él, debe cumplimentar los datos solicitados en los apartados 1, 2, 3, 4, 5, 6 y remitirnos, **sin cargo alguno**, la tarjeta a MARCOMBO, S.A.

2 ¿POR QUÉ ELIGIÓ ESTE LIBRO? — ELECCIÓN ❷

- Mejor tratamiento de la materia 1 ☐
- Conocimiento del autor 2 ☐
- Precio 3 ☐
- Otros 4 ☐

3 ¿CÓMO SE ENTERÓ DE LA EXISTENCIA DE ESTE LIBRO? — MEDIO ❸

- Ficha/Catálogo Marcombo 1 ☐
- Promoción por correo 2 ☐
- Anuncio prensa/radio/TV 3 ☐
- Recomendado por mi librero 4 ☐
- En una biblioteca 5 ☐
- Recomendado como texto 6 ☐
- Otros 7 ☐

4 ¿DÓNDE ENCONTRÓ ESTE LIBRO? — LUGAR ❹

- Librería 1 ☐
- Gran almacén 2 ☐
- Distribuidor 3 ☐
- Biblioteca 4 ☐
- Otros 5 ☐

5 AREAS DE SU INTERÉS — INTERÉS ❺

- Electrónica 1 ☐
- Informática 2 ☐
- Electricidad y Electrotecnia 3 ☐
- Automática/Control/Robótica 4 ☐
- Técnicas energéticas (Calor/Frío) 5 ☐
- Mecánica 6 ☐
- Automóviles 7 ☐
- Arquitectura y Construcción 8 ☐
- Matemáticas/Estadística 9 ☐
- Economía/Empresa 10 ☐

6 ❻

- GRADO DE SATISFACCIÓN POR LA LECTURA DE ESTE LIBRO (de 0 a 10) ☐☐
- ¿CUÁNTOS LIBROS DE NUESTRO FONDO POSEE? ☐☐☐
- SEGÚN USTED: ¿QUÉ TEMAS DEBERÍAN ABORDAR LOS FUTUROS LIBROS DE MARCOMBO? (Especifique los temas) TEMAS ▶

SERVICIO DE INFORMACIÓN PERIÓDICA

CÓDIGO ☐☐☐☐☐

❶

APELLIDOS ....

NOMBRE .... Tel. ....

Empresa ....

Domicilio ....

Marque si el domicilio es de la empresa ☐ o particular ☐

Población .... C.P. ....

Provincia ....

País ....

Firma

PARA UNA MEJOR INFORMACION NO OLVIDE CUMPLIMENTAR LOS DATOS DE LOS APARTADOS 1, 2, 3, 4, 5, 6.

____________, ____________

____________, ____________